TRACK LAYOUT ACCOMMODATING DYNAMIC ROUTING IN AUTOMATED MATERIAL HANDLING SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Junho Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Operations Research

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
August 2016

TRACK LAYOUT ACCOMMODATING DYNAMIC ROUTING IN AUTOMATED MATERIAL HANDLING SYSTEMS

Approved by:

Professor George Nemhauser,
Co-Advisor
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Dr. Dima Nazzal
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor Shabbir Ahmed, Co-Advisor
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor Byungsoo Na
Division of Business Administration
Korea University

Professor Joel Sokol, Co-Advisor
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Date Approved: April 25, 2016

ACKNOWLEDGEMENTS

First, I would like to thank Drs. George Nemhauser, Shabbir Ahmed, and Joel Sokol. Not only have I received a lot of academic lessons, but their support and advices also have made me continuously pursue the final goal.

Second, I appreciate for Samsung Electronics Corporation for the financial support, the internship opportunity, and the motivation for my research. Specifically, I had great experience to work with Dr. Byungsoo Na and Dr. Jee-Hyuk Park.

Third, thank you to my fellow students, especially Kelly Bartlett who solidified the foundation of my research. I also learned what I should do as an engineer from her. In addition, Tonghoon Suk and Hyunwoo Park have listened to my whining and cheered me up for six years.

Finally and the most importantly, thank you to my wife You Young and my parents. I believe their prayers wishing me to complete my goal are answered. It is now my turn to make my family happy. I cannot thank you enough.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . . **iii**

LIST OF TABLES . . . . . **vii**

LIST OF FIGURES . . . . . **viii**

SUMMARY . . . . . **x**

I INTRODUCTION **1**

1.1 System description 1

1.1.1 Motivation 1

1.1.2 Semiconductor manufacturing 2

1.1.3 Automated material handling systems 3

1.1.4 Transfer requests and vehicle routing 5

1.2 Problem description 6

1.2.1 Track layout accommodating dynamic routing 6

1.2.2 Shortcut placement problem 8

1.2.3 Challenges and contributions 8

II LITERATURE REVIEW **10**

2.1 Automated material handling systems in semiconductor manufacturing 10

2.1.1 Lot scheduling 10

2.1.2 Vehicle routing 11

2.1.3 Design problems for unified, segregated, and conveyor-based AMHSs 12

2.2 Alternative path routing 14

2.3 Network design problems 15

2.3.1 Network design problems with survivability 16

2.4 Integration of optimization and simulation 20

III NETWORK DESIGN PROBLEM WITH ALTERNATIVE PATHS 23

3.1 Introduction 23

3.2 Input data and assumptions 25
3.2.1 Input data 25
3.2.2 Assumptions on the base graph 25
3.2.3 Assumptions on commodities and alternative paths 29
3.3 Dynamic routing and alternative paths 29
3.4 Formulation 31
3.4.1 Decision variables 31
3.4.2 Constraints 31
3.4.3 Objective function 35
3.4.4 Additional approaches 40
3.5 Conclusions 41

IV INTEGRATION OF OPTIMIZATION AND SIMULATION FOR AN AMHS TRACK DESIGN 43

4.1 Introduction 43
4.2 Simulation 44
4.2.1 Model description 44
4.2.2 Simulation generator 45
4.3 Feedback from simulation to optimization 47
4.3.1 Commodities 47
4.3.2 Average travel time 48
4.3.3 Rerouting frequency 50
4.4 Integration of optimization and simulation 52

V COMPUTATIONAL RESULTS 56

5.1 Problem specifications 56
5.2 Base case results 60
5.2.1 Overview 60
5.2.2 Relationship between η and performance metrics 65
5.2.3 Characteristics of shortcuts in NDPA(b) 72
5.3 Sensitivity analysis 79

5.3.1 Sensitivity to routing schemes 79
5.3.2 Sensitivity to higher workload 82
VI CONCLUSIONS AND FUTURE WORKS 85
APPENDIX A — CONSTRAINTS FOR A BASE TRACK 87
REFERENCES 97

LIST OF TABLES

1 Path selection example 34

2 Bay-process assignments for 10-bay layouts 58

3 Bay-process assignments for 20-bay layouts 58

4 Layouts appearing multiple times in 10-bay instances 60

5 Layouts appearing multiple times in 20-bay instances 61

6 Comparison with other approaches 63

7 Performance metrics of a 10-bay instance (A01) 68

8 Paired t-test statistics of 10-bay instances (A01, time in system) . . . 69

9 Paired t-test statistics of 10-bay instances (A01, number of completed requests) 70

10 Predicted and actual shortcut directions 74

11 Frequency of deadlock 82

LIST OF FIGURES

1 An example of a clean room facility for semiconductor fabrication from www.engadget.com 2

2 An example of OHT vehicles from www.muratec-usa.com 4

3 An example track with 20 bays 4

4 Bad alternative path: low disjointed path 7

5 Bad alternative path: too costly path 7

6 An example spine layout with 20 bays 27

7 Simplification of bays in G ($BW = 3$, $BH = 3$) 27

8 Grid graph when $NB = 10$, $DC = 0$, $DO = 0$, $BW = 3$, $BH = 3$, and $BB = 1$ 28

9 No alternative path when $\gamma_1 \leq 10/14$ 30

10 An example of a base track 32

11 Multiple alternative paths and the value of AP 36

12 An example of multiple alternative paths 37

13 Combining optimization and simulation 43

14 Different speeds at the same location 51

15 Number of iterations (10-bay) 55

16 Number of iterations (20-bay) 55

17 Comparison with other routing approaches 64

18 Base case results (10-bay, dynamic) 66

19 Base case results (20-bay, dynamic) 67

20 Rerouting frequency (base, 10-bay) 71

21 Rerouting frequency (base, 20-bay) 72

22 Best layout of A01 73

23 Impacts of shortcuts on alternative paths 75

24 AP/SP vs. η 76

25 Delay ratio vs. AP/SP 77

26 Delay ratio vs. AP/SP (with 100 random designs) 77
27 AP/SP and performance metrics 78
28 Routing performance of static and semidynamic routing (10-bay) . . . 80
29 Routing performance of static and semidynamic routing (20-bay) . . . 81
30 Higher workload results (10-bay) 83
31 Higher workload results (20-bay) 84
32 Grid representation of bays 88

SUMMARY

Modern semiconductor fabrication facilities depend on automated material handling systems (AMHSs) to manage the processes and variations in time required to produce advanced semiconductor products. Unified AMHSs, which operate overhead hoist transport vehicles, support direct tool-to-tool transfers between different bays. AMHSs continue to use static routing, but dynamic routing schemes have been studied to improve traffic conditions. In this dissertation, we propose an approach integrating optimization and simulation to design a track layout of an AMHS employing dynamic routing. We present a network design problem accommodating alternative paths. Our formulation is based on a multi-commodity network design problem, and we add secondary flow variables to represent rerouted vehicles. The problem requires input data, especially a base graph and commodities, reflecting realistic traffic conditions. We use simulation to validate the design from the optimization problem and provide its input data. We obtain a solution design using a heuristic that combines optimization and simulation. Our computational results illustrate that the parameter that controls the uniqueness of alternative paths has a significant impact on the routing performance of vehicles.

CHAPTER I

INTRODUCTION

This dissertation proposes an optimization-simulation integration approach for designing the track layout of an automated material handling system in semiconductor manufacturing. The system operates overhead hoist transport vehicles that move work-in-process wafers. We formulate a network design problem to accommodate dynamic routing and employ simulation to incorporate realistic vehicle movements in the optimization problem. Our computational results illustrate that the parameter η, which controls the uniqueness of alternative paths, has a significant impact on the routing performance of vehicles. This chapter introduces the system, defines the problem, and discusses the challenges and contributions.

1.1 System description

1.1.1 Motivation

Semiconductor manufacturing affects almost every aspect of contemporary life. Laptops, tablets, and smartphones have largely replaced the use of desktops for many common applications, thanks to technological advances in display (from CRTs to LCDs), weight (from heavy televisions to portable visual devices), and resolution (from waves to pixels). The advances are possible because todays microchips are designed to hold billions of transistors. For example, in 1979, Intel 8086 held 29 thousand transistors, whereas in 2015, SPARC M7 held 10 billion transistors.

Because the global semiconductor manufacturing industry is highly competitive, manufacturers tend to focus on reducing the cost of materials, improving productivity, and building sustainable supply chains while still delivering well-designed, reliable end-products. Wafer fabrication is an expensive process and requires a variety of

machinery; some photolithography machines can cost billions of dollars. Each 300-mm wafer sheet costs about $4,000 to produce, and one well-known manufacturer produces more than 7 million wafers annually. The actual process requires clean room and storage facilities because wafers are sensitive to external stimuli and are easily contaminated by airborne particles. Figure 1 illustrates a typical wafer fabrication facility, which we abbreviate to a fab in this dissertation.

Figure 1: An example of a clean room facility for semiconductor fabrication from www.engadget.com

1.1.2 Semiconductor manufacturing

Semiconductor manufacturing requires numerous processing steps with re-entrant flows. Depending on the product recipe, wafers will return to the same machine multiple times. For example, wafers need to visit photolithography machines multiple times in order to imprint the required circuits. Since wafers are sensitive to external circumstances and vulnerable to contamination, they are inspected throughout fabrication. Wafers that fail an inspection will be reworked. Thus, wafer fabrication

is a multiple-machine/multiple-wafer process where wafers at different stages of the processing sequence compete for the same resource.

Developing an optimal manufacturing schedule is not a trivial task. In general, semiconductor products require more than 400 processing steps and several weeks until completion. Variability in processing time also complicates optimal scheduling. Wafers in the same manufacturing step can require different processing times at the same machine, even though the purity of wafer ingots is highly controlled.

1.1.3 Automated material handling systems

Inefficient material handling can cause unexpected idle time of machines and decrease productivity. Therefore, modern fabs depend on automated material handling systems (AMHSs) to manage the processes and variations in time required to produce advanced semiconductor products. Before material handling became automated, wafers were carried manually in a cartridge, such as a front opening unified pod (FOUP), because of their extremely sensitive nature. However, as the diameter of a wafer increases, cartridges became too heavy for manual handling. For example, 25 pieces of 300-mm wafer sheets in a FOUP cartridge weigh 20 pounds [5]. Consequently, AMHSs have been widely adopted in semiconductor manufacturing.

AMHSs operate various transportation modes, such as automated guided vehicles, conveyor belts, and overhead hoist transport (OHT) vehicles. In this dissertation, "vehicles" denotes OHT vehicles unless specified otherwise. Figure 2 shows OHT vehicles moving along the track located on the ceiling of a fab. The OHT vehicles extend stretchable arms to pick up or drop off FOUPs below (they also secure the FOUPs inside before moving again). Picking up a FOUP from a machine or a storage location and dropping off a FOUP to a machine or a storage location are called loading and unloading, respectively.

In most fabs, AMHS track follows a spine layout. The tracks are unidirectional

Figure 2: An example of OHT vehicles from www.muratec-usa.com

and the intersections are of two types: merging or diverging. There are no parallel tracks between two locations. Multiple loop tracks, which we call a center loop, are in the middle. Bays, which are the locations where machines with the same or similar functions are grouped, are located along the center loop. In the bays, vehicles stop above machine ports or side-track-buffers (STBs) for loading or unloading. All of the loading and unloading locations of a bay are connected. Some fabs have large loop tracks surrounding their bays, which we call an outer loop. Figure 3 shows a spine layout with 20 bays of the same size, 4 parallel lanes in the center loop, and 2 parallel lanes in the outer loop.

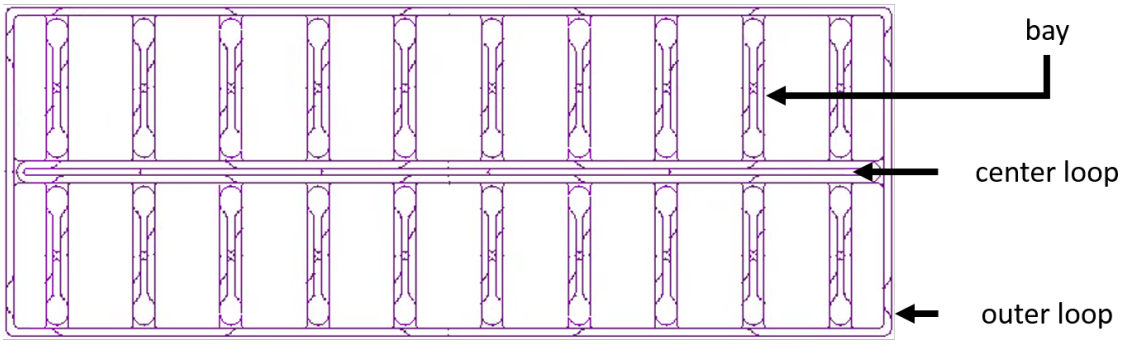


Figure 3: An example track with 20 bays

AMHSs with OHT vehicles are of two types: unified or segregated. Unified

AMHSs support direct tool-to-tool transfers between different bays, whereas segregated AMHSs have bay stockers connecting bays and the center loop and the vehicles travel either inside one bay or along the center loop. Bay-to-bay transfers are split into in-bay (origin), center loop, and in-bay (destination) transfers. We are interested in unified AMHSs because even though it is the predominant system in fabs, unified AMHSs have been less studied.

1.1.4 Transfer requests and vehicle routing

Both types of AMHSs generate transfer requests as needed. When a machine completes a job and releases a FOUP, the system sends another FOUP to the machine. If multiple FOUPs are waiting for the same machine, the system selects one of them based on a predefined prioritization rule. On the other hand, the system also decides the storage location for the released FOUP using another rule. Unified AMHSs also support direct tool-to-tool delivery, which sends the released FOUP directly to another machine. When a direct delivery is made, the system also prioritizes the destination machines. Some academic studies have proposed an exact solution approach based on disjunctive graph representations, but in practice most fabs use selection rules.

The AMHS assigns vehicles to transfer requests for FOUP pickup and delivery and selects the paths for the two trips, i.e., to the origin and to the destination. The origin and destination can be either a machine or a storage location. When the assigned vehicle arrives at the origin, it picks up the FOUP and moves to the destination where it drops off the FOUP to complete the transfer request. The vehicle becomes idle until it is assigned to another transfer request.

To the best of our knowledge, AMHSs in fabs continue to use static routing for dispatching assigned vehicles. Static algorithms, however, cannot reflect current traffic conditions, and thus are vulnerable to congestion because vehicles assigned to the

same transfers take the same path. To alleviate congestion, fab operators can update long-term congestion information on the system or execute manual interventions [60], but these are temporary remedies since static routing cannot immediately respond to instant congestion.

On the other hand, dynamic routing incorporates up-to-date traffic conditions and recalculates the fastest path between two locations. Bartlett et al. [16] proposed a dynamic routing scheme called "learn-and-adapt" that estimates travel time as the exponentially weighted average of historical data and assigns the location when a vehicle approaches a diverging intersection where it can change its path. Because of congestion, the selected path is not necessarily the shortest path. In this dissertation, when dynamic routing is considered, the term "uncongested shortest path" denotes the traditional shortest path which minimizes the total distance. We use the term "shortest path" to denote the fastest path based on current traffic information.

1.2 Problem description

1.2.1 Track layout accommodating dynamic routing

We propose a network design problem that accommodates vehicle movements under dynamic routing, and examine whether a specific track layout improves the performance of dynamic routing. Dynamic routing dispatches vehicles via various paths based on traffic conditions and tells vehicles to change paths while they are moving, which is called rerouting. Dynamic routing does not necessarily use all paths to reroute vehicles. Figure 4 illustrates a path that shares too many track segments with the uncongested shortest path. In addition, Figure 5 shows a path that is quite longer than the uncongested shortest path. Hence, a good alternative is a path which does not share many track segments with the shortest path, and is not too costly. As long as we can control the cost of the uncongested shortest paths, providing alternative paths may result in more effective rerouting and eventually reduce congestion.

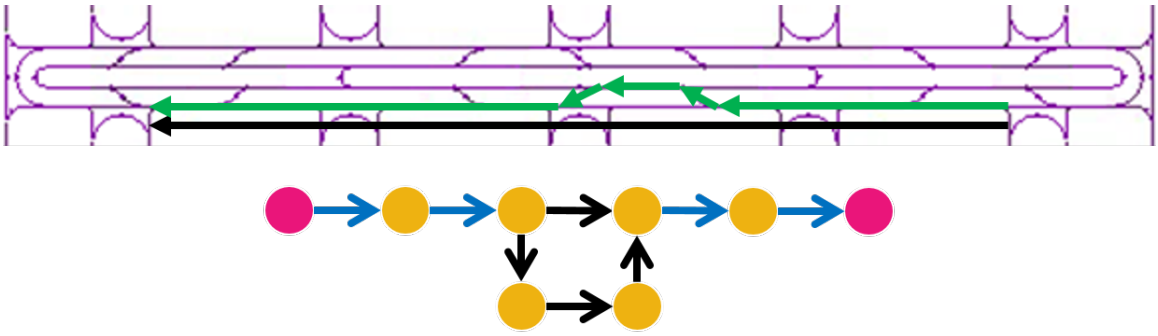


Figure 4: Bad alternative path: low disjointed path

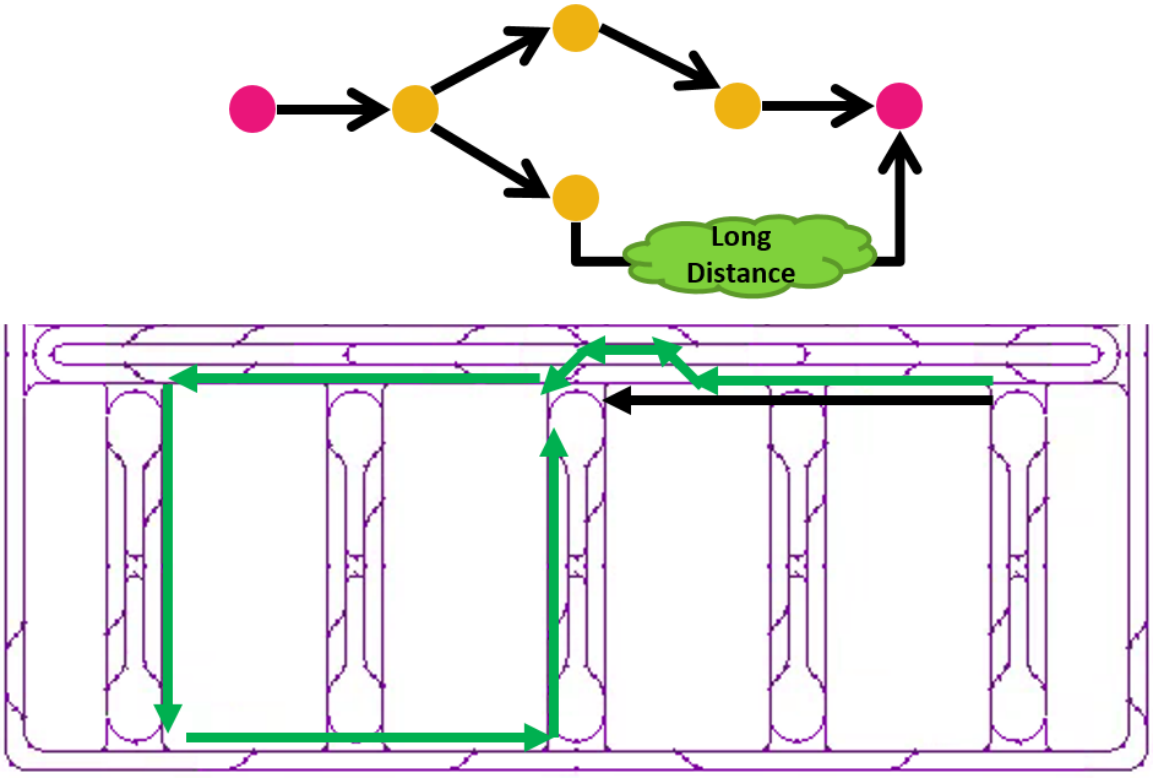


Figure 5: Bad alternative path: too costly path

Accordingly, we aim to find a track design with useful alternative paths. Specifically, in Chapter 3, we define a network design problem accommodating alternative paths. We modify the definition of an alternative path found in studies of vehicle navigation systems and apply it to our design problem.

1.2.2 Shortcut placement problem

Designing the entire track layout of a fab AMHS is a rare event. First, building a new facility requires significant investments of money and time, e.g., one industry leader invested $10.2 billion to construct a new fab, which took 1.5 years to complete. Second, modifying the entire track layout of an existing fab is impractical due to the negative impacts on vehicle operations and productivity. The solution is to undertake design adjustments, such as changing or adding shortcuts, which are less disruptive.

In this dissertation, "shortcuts" denote the track segments connecting two lanes of the center or outer loop. They are the only way to distribute traffic because adjacent locations are connected by single lanes. The shortest paths of origin-destination pairs may contain several shortcuts. In addition, vehicles take shortcuts when they are rerouted. Hence, the impact of dynamic routing and shortcuts are interrelated. To reflect the practical aspect of AMHS installation and the importance of shortcuts, we build a spine layout based on operational and physical input data. Then we select the locations and directions of shortcuts in the center and outer loops.

1.2.3 Challenges and contributions

Although we hypothesize the following shortcut placement problem, we face two immediate challenges because shortcut placement involves decision making in a discrete solution space. The first challenge is to designate the location of shortcuts, given the fab size. Our shortcut placement problem assumes that a fab with 10 bays has 45 candidate locations (actual fabs are much larger and have more candidate shortcut locations). There are 3^{18} possible selections only for the outer loop, and there are more than 3^{20} selections for the center loop. Hence, the number of feasible shortcut placements is more than $3^{38} \approx 10^{18}$. We will propose a network design problem for selecting shortcuts. The second challenge is how to fill the gap between a static layout design and dynamic traffic conditions. Solving the problem requires assembling data

on travel time and the usage of alternative paths; both are the outcome of a track layout and not its input. High-fidelity simulation can provide the data, but we know that combining a track design problem and simulation introduces other issues. We will process the simulation results to obtain input data of optimization and define a procedure to obtain a track design from the combination.

In this dissertation, we propose an approach integrating optimization and simulation to design a track layout. In Chapter 2, we review the relevant published literature on AMHSs, network design problems, and the integration of optimization and simulation. In Chapter 3, we present a network design problem accommodating alternative paths. Our formulation is based on a multi-commodity network design problem, and we add secondary flow variables to mimic rerouted vehicles. The problem requires input data reflecting realistic traffic conditions. A base graph and commodities depend on the simulation configuration, so we define them a priori. Travel time and rerouting data are constructed based on simulation. In Chapter 4, we present the calculations for travel time and rerouting data and explain how to obtain a track design from the iterations of optimization simulation. In Chapter 5, we give the computational results of our design solution compared with the two approaches and a classic multi-commodity network design problem presented in Chapter 3. We report the significant relationship between routing performance and a parameter in our problem. In Chapter 6, we conclude and give suggestions for extending the research in this dissertation.

CHAPTER II

LITERATURE REVIEW

The following sections survey the published literature relevant to this dissertation. First, we review the literature on AMHSs in semiconductor manufacturing including lot scheduling, vehicle routing, and design problems. Second, we review the literature on survivable network design problems in addition to classic design problems. The chapter ends with an examination of the key studies on integration of optimization and simulation.

2.1 Automated material handling systems in semiconductor manufacturing

Agrawal and Heragu [5] presented a comprehensive survey of industry characteristics, structure of manufacturing systems, AMHSs, facility layout designs, and other aspects. Their survey focused on segregated and conveyor-based AMHSs, rather than unified AMHSs. In addition, many studies about AMHSs have been published in the semiconductor industry, and we review the literature on three topics: lot scheduling, vehicle routing, and design problems.

2.1.1 Lot scheduling

Lot scheduling is a classic research topic in semiconductor manufacturing, which is job shop scheduling with unique characteristics. Recently, Xie and Allen [104] provided a comprehensive survey of job shop scheduling problems applied to material handling systems.

In semiconductor manufacturing, scheduling relies on dispatching rules, e.g., wafer lots waiting for the same machine or machines waiting for the next wafer lot are

prioritized according to local (processing time, due date) or global (average work-in-process level) characteristics, respectively. Lu et al. [62] defined several prioritization rules, each of which was expected to improve a different performance metric. A dispatching rule proposed by Li et al. [61] aimed to balance machine utilization.

Scheduling in semiconductor manufacturing has online characteristics, which means that dispatching rules do not perform as expected due to complexity in manufacturing. Mittler and Schoemig [67] built industry-scale simulation models to compare the performance of the dispatching rules proposed by Lu et al. [62] and Li et al. [61]. They observed that dispatching rules, especially the least slack method in [62], had unexpected impacts on performance metrics. Rose [82] showed that the shortest processing time first rule was not always optimal in complex job shop scheduling. Rose [83] also found that prioritization based on the critical ratio of a job (the ratio of its remaining time until the due date to its total remaining processing time) was affected by the assumptions regarding waiting and transfer time in manufacturing.

Combining multiple rules has also been studied. In Dabbas et al. [32] and Dabbas and Fowler [33], each dispatching rule assigned a score to each job, and jobs were prioritized based on a linear combination of the scores. Tyan et al. [93] and Chen et al. [24] suggested a method that applied different dispatching rules depending on the circumstance. Kuhl and Laubisch [57] combined rework strategies and dispatching rules, and Wu et al. [102] studied coupling of policies for dedicated and non-dedicated machines.

2.1.2 Vehicle routing

Three important topics in vehicle operations are vehicle assignment, path selection, and deadlock prevention. Kim et al. [52] proposed a scheme to reassign vehicles to transfer requests. Kim et al. [54] presented a vehicle assignment algorithm based on the Hungarian algorithm, which produced better results than the algorithm in [52].

Both papers compared routing performance metrics with the assignment algorithms introduced by Bozer and Yen [19] and Le-Anh and de Koster [59].

The AMHS literature on dynamic routing is scarce. Patents by Gaskins et al. [39] and Huang et al. [47] proposed dynamic path selection based on current traffic conditions, but neither considered congestion avoidance by rerouting vehicles to different paths. To the best of our knowledge, Yang et al. [105] first suggested a dynamic routing algorithm based on the number of vehicles on a path. Their computational results illustrated the superiority of dynamic routing over static routing. Bartlett et al. [16] proposed a scheme that predicted travel time by exponential smoothing of historical data. Their algorithms produced significant improvements on routing performance even when heavy congestion, deadlock, or vehicle breakdowns occurred or when the system was in a steady-state.

Im et al. [48], who defined three categories, topology, capacity, and power supply, based on source, recommended applying a specialized approach depending on category. Their waiting relation matrix indicated the locational relationship between every pair of vehicles. Since each category of deadlock had unique characteristics in a waiting relation matrix, the authors suggested using the matrix as an efficient method for deadlock detection. They provided practical deadlock examples to demonstrate its application. Kim et al. [53] suggested a simple approach to prevent vehicle blocking among vehicles traveling to pick up a wafer lot. Deadlock prevention has been a topic in the literature on automated guided vehicles and flexible manufacturing systems ([103], [25], [68], [101]).

2.1.3 Design problems for unified, segregated, and conveyor-based AMHSs

Unified AMHSs are relatively new technologies and the related literature is in its infancy. Kurosaki et al. [58] predicted that an AMHS that could support direct delivery similar to a unified AMHS would outperform segregated AMHSs. The authors

proposed a track layout, a mixture of segregated and unified systems, and compared routing performance. They concluded that a mixed system was infeasible. Bahri et al. [11], who compared four possible configurations of AMHSs (segregated, unified, and their combinations), found that unified AMHSs reduced average travel time by 32% compared with segregated systems. The authors mentioned that direct delivery in unified AMHSs required more attention when designing a track layout and routing vehicles. They also stated that the specifications of unified AMHSs had been mentioned in the 1999 International Technology Roadmap for Semiconductors, which is the first relevant publication to the best of our knowledge. Sturm et al. [90] compared two layouts via simulation based on traditional performance metrics. Both layouts differed from spine layouts. None of the aforementioned papers explained how to build the layouts.

To date, most track layout design problems are designed for segregated AMHSs. Yang and Peters [106] formulated a layout design problem as a second-order cone program with a linearization technique. Their example track layout had a center loop with a single lane and multiple shortcuts. The authors did not apply their linearization approach, however, and used a commercial solver to solve the problem. Hsieh et al. [46] designed a track layout with a partially bidirectional loop consisting of two unidirectional tracks. Their layout reduced average cycle time and increased stocker utilization.

Although Nazzal and Bodner [69] distinguished segregated and unified AMHSs in their design framework, we found no subsequent research with concrete examples of unified AMHSs. Chung and Tanchoco [27] studied a layout design problem for unified AMHSs. They assumed several polygons and divided them into multiple segments to evaluate the average travel distances for pairs of polygon segments. They developed simulation models for one hexagonal and two rectangular layouts all with 36 machines. The hexagonal layout had shorter average travel time than the rectangular layouts,

but its feasibility was questionable unless validated by larger layouts and realistic traffic information. Using a simulation model of a real-size fab, Bartlett [15] compared the routing performance of four track designs of a unified system with dynamic routing. It was observed that routing performance varied significantly according to track layout, but finding the best layout was not determined.

Conveyor-based AMHSs, another type of AMHS, are popular in LCD manufacturing. Work-in-process LCD panels are too large for transport by OHT vehicles, but are robust to vibration. Nazzal and El-Nashar [70] surveyed conveyor-based AMHSs, and Nazzal et al. [71] proposed an analytical performance model for layout design problems. Wang [97], who assumed that conveyor systems would be suitable for fab manufacturing 450-mm wafers, formulated a facility design problem.

Identifying the relationship between routing performance and layout characteristics assists in designing track layouts. Nazzal and McGinnis [72], [73] studied vehicles on the center loop in segregated AMHSs. They used target machine, job type, and current status to define the state of a vehicle. They built a Markov chain to represent state transitions and estimated routing performance. Govind et al. [42] built a closed queuing network model to approximate a segregated AMHS. Mackulak and Savory [64] compared two track designs with different storage locations with respect to average travel time, storage utilization, and number of deliveries. They recommended distributed storage systems, which many fab AMHSs employ. We note that it is difficult to predict routing performance in unified AMHSs. Inter-bay transfers require that vehicles travel longer distances than segregated systems, which can cause more complicated interactions among vehicles.

2.2 Alternative path routing

Alternative path routing has received attention in the literature due to the development of vehicle navigation systems in internet-based map services. The suggested

routes are based on user requirements and current traffic conditions. The shortest path with respect to distance may not be the fastest path in crowded areas; instead, the second- or third-shortest path may provide the fastest trip. Users can ask for paths that avoid highways or toll roads.

Bader et al. [10] studied alternative path routing from the perspective of drivers. They proposed attributes to analyze alternative paths and showed NP-hardness of optimizing any attribute pair. The authors also introduced several heuristics and checked if alternative paths made sense to drivers. They conducted a user survey to evaluate the computational results from their heuristics. The penalty method mentioned in [10] was studied more extensively by Kobitzsch et al. [56]. Abraham et al. [1] developed an algorithm to compute reasonable alternative paths defined by the three characteristics described in Chapter 3. They defined a via-path through node v, which was a concatenation of the two shortest paths from source node s to v and from v to terminal node t. Their algorithm based on bidirectional Dijkstra checked for every node v, if the via path through v satisfied the three characteristics. Kobitzsch [56] used the same definition of alternative paths in Abraham et al. [1], but reduced the problem size by focusing on more viable candidate paths. Apparently, the algorithm reduced runtime although its success rate, the frequency of finding an admissible path in test instances, was not much different from the algorithm in [1].

2.3 Network design problems

A network design problem searches for a subgraph of a given graph, which satisfies predefined requirements. Subgraphs may provide the shortest path between any two locations or secure sufficient connectivity of multiple commodities. Network design has been applied to transportation, facility location, wireless communications, power systems, water resource planning, and other areas. Magnanti and Wong [66] provided a comprehensive survey of network design problems and solution algorithms. They

categorized network design problems according to commodities, objective functions, and additional constraints. Costa [29] surveyed the literature on fixed-charge network design problems. Problem types were defined based on capacities, commodities (single, multiple), technologies for sending commodities, and cost functions, and relevant papers were cited for each problem type.

A variety of real-world applications of network design problems have been studied. Gavish [40] introduced the Telepak problem, which decided the structure and maximum traffic volumes of a local access network. Crainic and Laporte [31] and Crainic [30] defined three levels of transportation planning and provided network design models. A long-term planning problem was modeled as a capacitated network design problem. Mid- and short-term planning problems were formulated as vehicle routing problems. Sherali and Smith [87], who studied a network design problem for water distribution, applied a reformulation-linearization technique to obtain the lower bound. Binato et al. [18] studied a capacitated network design problem for power transmission networks. Their approach based on Benders decomposition reduced CPU time in the numerical results for the southeast Brazilian power system. Randazzo and Luna [79] formulated a local access network design as an uncapacitated problem with single commodity and extended it to a multi-commodity formulation. They described branch-and-bound, branch-and-cut, and Benders decomposition algorithms and compared the computational results for the test problems.

2.3.1 Network design problems with survivability

Real-world networks are vulnerable to malfunctions caused by many types of hardware failures. Disconnected optical fiber cables, overloaded power lines, and congested road networks are examples of edge failures. Routers turned off due to insufficient power supply, and maximum lifespans of electrical or communication devices that depend on regular and peak work load levels are examples of node failures. Network failures and

subsequent "cascades" can cause widespread disasters. The power blackouts in North America in 2003 and in India in 2012 affected 55 and 620 million people, respectively, in only two days. A ships anchor striking an undersea optic cable in 2012 resulted in a 20% slowdown in internet access in several East African countries.

Researchers have proposed various methods to define survivability of a network. For example, Jan [81] assumed that every component had the same failure probability and defined survivability as their aggregation. Optimal solutions were found when a base graph $G = (N, E)$ satisfied $|E| = |N| - 1$, $|E| = |N| + 1$, or $|E| = |N|$. The author reported computational results with $|E| \leq 23$.

Graph-theoretic measures provide an alternative definition. The connectivity of a graph is defined as the number of edge- or node-disjoint paths for each node pair. If there are k edge-disjoint paths for each commodity, a network is still functional with any $k - 1$ edge failures. Functionality after a network failure is also important. A spare network is defined as a subgraph after node or edge failures occur. Network restoration problems and network interdiction problems study sequential survivability based on spare networks. Raghavans dissertation [77] and two co-authored papers [78], [65] presented network design problems with connectivity requirements. In [78], linear-time algorithms were presented for series-parallel graphs with low connectivity, a popular subgraph in telecommunication networks. In [65], a framework for finding stronger formulations was presented.

Grötschel et al. [43] examined a variety of network design problems with survivability. Focusing on minimum spanning tree, Steiner tree, and minimum cost k-connected network design problems, the authors studied their structural properties and the facet-defining inequalities of integer programming formulations, and proposed heuristics with computational results. They also described polynomially solvable cases. Kerivin and Mahjoub [50] surveyed the literature on survivable network design problems; their survey cited papers that presented polynomially solvable cases,

heuristics, approximation algorithms, and polyhedral characteristics. Dahl and Stoer [34] described a special case of survivable network design problems where survivability was imposed on a given set of commodities. Assuming that edge capacities were step increasing, they formulated the problem using band inequalities and presented a cutting-plane algorithm. The computational results showed that flows split without network failures. Soni and Pirkul [89] formulated a straightforward formulation for a survivable network design problem. To obtain Q edge-disjoint paths, they fixed the volume of each commodity to Q and constrained the maximum volume of each commodity of an edge to less than or equal to 1. They provided a decomposition algorithm that found cuts from minimum spanning tree problems. Balakrishnan et al. [14] proposed a survivable network design problem with connectivity requirements. The types of inequalities, connectivity upgrading inequalities, generalized forcing inequalities, and design inequalities they proposed were expected to strengthen the formulation.

Network restoration consists of line restoration and path restoration. When an edge failure occurs, line restoration searches for an alternative path that connects the flow on the failed edge, and path restoration provides path substitutes from origin to destination of the disconnected commodities. Kennington et al. [49] defined three types of survivable network design problems: working capacity allocation, spare capacity allocation, and a joint planning model. Sakauchi et al. [84] considered path restoration as line restoration with multiple edge failures. They provided a linear programming formulation minimizing the number of spare channels. They provided a linear programming formulation minimizing the number of spare channels. Grover et al. [44] utilized edge-disjoint paths for line restoration. Balakrishnan et al. [13] and Balakrishnan et al. [12] studied line restoration problems with single and multiple types of facilities. The types were characterized by the capacity and cost of

each component. In [13], they studied polyhedral characteristics and provided facet-defining inequalities and in [12], they provided heuristics and worst-case analyses. Chujo et al. [26] proposed a heuristic for a path restoration problem. They assigned initial spare capacities based on the shortest path, and the additional assignment on alternative paths was imposed until the goal was achieved. Because path restoration considers the entire path affected by a network failure, it is less costly than line restoration, as mentioned by Veerasamy et al. [96]. Agarwal [3] studied network restoration using the approach presented in [4] where the node set was partitioned into k-partitions, and each partition was treated as a single super node.

Network interdiction problems which assume intentional attacks on a network have attracted researchers in military and homeland security applications. Wollmer [99] presented research based on the max-flow min-cut theorem. Cormican et al. [28] studied stochastic network interdiction problems. A binary random variable determined the success of an enemys attack on a network component. In Wood [100], an interdiction problem against an illegal drug supply network was studied. Smith et al. [88] proposed a network interdiction problem where the enemy, which had a budget constraint and a multi-purpose attack, attempted to disconnect the edges with the largest initial flows or capacities, or remove the edges that caused the greatest loss. The initial design problem was formulated as a mixed-integer program (MIP) maximizing the weighted sum of flows before and after interdiction. The problem was then incorporated inside a bi-level problem for each type of interdiction.

There are two differences between the problem discussed in this dissertation and survivable network design problems. First, the alternative paths in our problem need not be completely edge-disjoint; rather, we focus on "partially" disjoint paths. The origin nodes have one outgoing edge and the destination nodes have one incoming edge. Hence, for each commodity, every path shares two edges (the outgoing edge from origin and the incoming edge to destination). Partially disjoint paths have

been studied in telecommunication networks, but most papers focus on routing ([94], [95], [21], [107]). Second, any congestion in our problem does not block traffic flows completely, but it does increase travel time. Vehicles in an AMHS move on unidirectional tracks, i.e., they cannot move when another vehicle is out of order or a deadlock occurs. Since such incidents are unpredictable and rare, incorporating them into a network design problem results in unnecessary conservativeness. Hence, we allow every vehicle to use any edges selected by the design variables while noting that the prevailing traffic conditions determine whether an alternative path uses a certain edge.

2.4 Integration of optimization and simulation

In our study, we combine optimization and simulation to design a track layout of a fab AMHS. Optimization and simulation have their own strengths and limitations, and integrating them to combine the strengths of both sides has been a classic research topic. In 1972, Nolan and Soverign [75] developed an iterative approach using optimization and simulation to design a transportation network, and in 1983, Shanthikumar and Sargent [86] classified combination (hybrid) approaches based on the relationship between optimization and simulation. The test instances in both studies were too small to apply those approaches to other areas. However, huge leaps in computing power since mid-1980s has promoted extensive studies in combining optimization and simulation.

Recently, Figueira and Almada-Lobo [37] surveyed the literature regarding integrating optimization and simulation. They reviewed iterative approaches as well as well-known simulation-optimization (SO) approaches. Carson and Maria [22], Andradóttir [7], Azadivar [9], Swisher et al. [92], April et al. [8], Fu et al. [38], and others described basic concepts, methods, and techniques of SO. A recent survey on SO was presented by Wang and Shi [98]. Although most SO approaches guarantee

convergence, they become computationally expensive when simulation is high-fidelity. The iterative approaches cited in [37] aimed to resolve this issue.

These iterative approaches have been applied to a variety of research problems including supply chain configuration, production planning, and water resource distribution. Chandra and Grabis [23] discussed hybrid models combining optimization and simulation for supply chain configuration. They classified hybrid approaches into sequential and simultaneous approaches, introduced their differences, and cited relevant literature. They emphasized the importance of two issues: what simulation provides to optimization and when the iteration between optimization and simulation terminates. Acar et al. [2] proposed a hybrid approach to solve a multi-period multi-product facility location problem with uncertainty. They updated the lower bound of the optimization problem based on the simulation results. Almeder et al. [6] applied an iterative framework between optimization and simulation to a supply chain network application. They formulated an MIP to obtain policies for simulation and updated the optimization problem based on the simulation results. They observed gradual decreases of the gap of total cost between optimization and simulation. Keizer et al. [35] designed logistics networks for perishable products like flowers. Their MIP determines hub placements and product deliveries with quality constraints. Their simulation model verifies the design obtained from the optimization. If the design is infeasible, then it updates the quality constraints so that the optimization provides a new design. Otherwise, i.e., when the design is feasible, the iteration terminates. Sel and Bilgen [85] and Bilgen and Çelebi [17] integrated decision making in production and distribution of soft drink and dairy products, respectively. Both studies employed MIP formulations to find production scheduling, updated the formulations using capacity constraints based on simulation results, and terminated the iteration when feasible scheduling is found. Byrne and Bakir [20], Kim and Kim [51], and Gnoni et al. [41] developed hybrid approaches for production planning problems. They also

employed simulation models to find capacity constraints for optimization. Sun et al. [91] proposed an iterative approach for designing a water supply network. Their formulation was a QP and the simulation updated the input data, such as tank volume, of optimization. Rani and Moreira [80], who surveyed the literature on reservoir system operations, cited papers on iterative approaches between optimization and simulation. Nguyen et al. [74] surveyed simulation-based optimization approaches to analyze the energy consumption efficiency of commercial buildings. Henderson and Mason [45] presented an iterative algorithm to solve a rostering problem. Their approach used simulation to find a cut for their optimization problem, but they did not present experimental results.

Simulation is essential for semiconductor manufacturing because of its complexity, so researchers have developed high-fidelity simulation models to test their approaches. Nevertheless, to the best of our knowledge, no studies analyzed their simulation results to update their analytic models. For example, Yang and Peters [106] solved a track design problem based on a predefined data set and compared routing performance using simulation. The optimization problem required edge distances but nothing from simulation results.

CHAPTER III

NETWORK DESIGN PROBLEM WITH ALTERNATIVE PATHS

3.1 Introduction

In this chapter, we formulate a network design problem (NDP) accommodating alternative paths. Our formulation is a special multi-commodity NDP. Given a base graph, NDP searches for a subgraph which satisfies predefined properties. One of its variants, a multi-commodity NDP requires a feasible subgraph to guarantee that each commodity has a flow from its origin to its destination. We let $G = (N, E)$ and K be a base graph and a set of commodities; commodity k specifies its origin o_k, destination d_k, and demand w_k. $y \in \{0,1\}^{|E|}$ and $x \in [0,1]^{|E|\times|K|}$ are the vectors of the design and flow variables, respectively. They have cost vectors: $f \in \mathbb{R}_+^{|E|}$ for y and $c \in \mathbb{R}_+^{|E|\times|K|}$ for x. In general, the objective function is to minimize the total cost, and the formulation of the multi-commodity NDP is:

$$
\begin{aligned}
\text{(NDP):}\quad \min\quad & \sum_{e\in E} f_e y_e + \sum_{k\in K}\sum_{e\in E} c_e^k x_e^k \\
\text{sub. to}\quad & \sum_{e\in\delta^+(i)} x_e^k - \sum_{e\in\delta^-(i)} x_e^k = \begin{cases} w_k & \text{if } i = o_k, \\ -w_k & \text{if } i = d_k, \\ 0 & \text{otherwise,} \end{cases} && \forall k \in K,\ \forall i \in N, \\
& x_e^k \le w_k y_e && \forall k \in K,\ \forall e \in E, \\
& y \in \mathcal{Y},\ x \in \mathcal{X}(y), \\
& y_e \in \{0,1\} && \forall e \in E, \\
& x_e^k \ge 0 && \forall k \in K,\ \forall e \in E,
\end{aligned}
$$

where $\delta^+(i)$ and $\delta^-(i)$ denote the sets of edges whose tails and heads are $i \in N$, respectively. Using $\mathcal{Y}$ and $\mathcal{X}(y)$, we can impose additional constraints on feasible subgraphs and flows. If c_e^k equals the time to traverse edge e, then NDP searches for a subgraph where the weighted sum of the shortest paths of commodities is minimized.

Commodities reflect traffic conditions in a fab AMHS. We build commodities based on the method to generate transfer requests in an AMHS simulator. To reduce the problem size, we combine the commodities according to their origin and destination locations. The demand of each commodity depends on how often the corresponding transfer requests are generated. Vehicles can travel along a path that is not the shortest, so we define additional flow variables corresponding to the traffic over alternative paths. Moreover, we allow the alternative path flow to be separated into multiple fractional flows to reflect that the AMHS sends vehicles along multiple paths based on traffic conditions. To do this, we impose additional decision variables and constraints and control the uniqueness of alternative paths compared to the shortest path using a parameter. Finally, we minimize the convex combination of the costs of the two flow variables.

Recalling that dynamic routing dispatches vehicles along multiple paths based on congestion, we incorporate multiple alternative paths and assume a spine layout with bays along center loop tracks which are surrounded by outer loop tracks. We also construct a grid graph containing every feasible design as a subgraph. In addition, our formulation includes how frequently the system reroutes transfer requests with the same origin-destination pair, to represent the importance of alternative paths.

In Section 3.2, we introduce the input data and assumptions of our problem. In Section 3.3, we review the definition of an alternative path in the literature on vehicle navigation systems and modify the definition based on the conditions of our problem. In Section 3.4, we formulate the problem as a mixed integer program. Section 3.5 concludes.

3.2 Input data and assumptions

3.2.1 Input data

Our problem has the following input data:

- $G = (N, E)$ is a directed graph that contains every feasible design as a subgraph. N and E are the sets of nodes and edges, respectively.
- K is a set of commodities; each commodity $k \in K$ is defined by its origin o_k, destination d_k, and demand $w_k = w_{o_k,d_k}$.
- $c \in \mathbb{R}_+^{|E|}$ is a cost vector of edges. We assume that c_e represents the average travel time on edge e.
- $\lambda \in [0, 1]^{|K|}$ is a vector of weights for the commodities, each of which reflects the relative frequency or probability of not using the shortest path of commodity k.

The first three items are necessary for a general multi-commodity network design problems while the fourth one is special to our problem. In our problem, $G = (N, E)$ depends on the size of a track layout, and K depends on the production information as well as the track size. We specify a lower bound on the uniqueness that all candidate alternative paths must satisfy. The parameter $\eta \in [0, 1]$ is the maximum portion of the edges on the shortest path, which are shared by alternative paths. For example, if an alternative path shares 80% of edges of the shortest path, then it is not feasible when $\eta < 0.8$.

3.2.2 Assumptions on the base graph

We assume that the base graph G is a grid graph with alternating directed edges between adjacent nodes. It should be able to contain all feasible track layouts but should not be excessively large. We designate intersections and candidate locations of shortcut tails or heads as nodes, and the other nodes maintain a grid graph. The edges

are track segments connecting two adjacent nodes. We exclude stopping locations, such as machine ports and storage locations, from N because our design problem identifies the aggregated flow of vehicles, not the movement of a specific vehicle. This assumption also corresponds to the dynamic routing scheme in our simulation model.

The following values define the size of a track layout. The last three do not represent the actual size of bays.

- NB: number of bays
- NC: number of (additional) lanes in the center loop (≥ 1)
- DC: 1 if the directions of lanes in the center loop alternates; 0 if they are same
- NO: number of (additional) lanes in the outer loop (≥ 1)
- DO: 1 if the directions of lanes in the outer loop alternates; 0 if they are same
- BW: number of horizontal nodes to represent a bay
- BH: number of vertical nodes to represent a bay
- BB: number of nodes between a bay

We do not count the innermost lane of the outer loop and the outermost lane of the center loop, which are connected to bays. For example, the track layout in Figure 6 has four lanes in the center loop and two lanes in the outer loop, but we assume that $NC = NO = 1$.

We simplify bays to squares of $BW \times BH$ nodes as illustrated in Figure 7. BW is related to the maximum number of shortcuts placed on the track segment between the entrance and exit nodes of a bay. When $BW = 3$, two outer nodes are the entrance and exit nodes of a bay, and the middle node can be the tail or head of a shortcut edge. BH relates to the number of shortcuts in the left and right tracks on the outer loop. BB is similar to BW: the maximum number of shortcuts placed

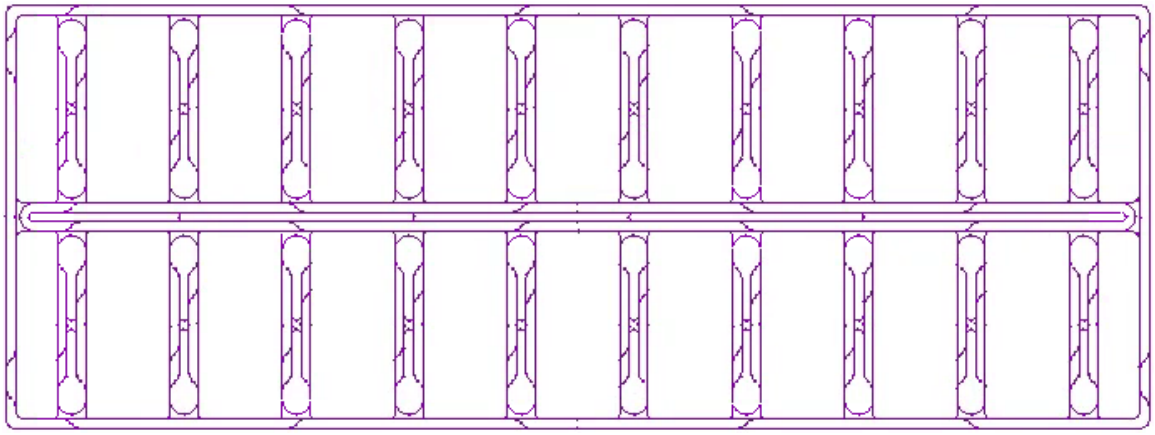


Figure 6: An example spine layout with 20 bays

between two bays equals BB. We partition the bays into one group above the center loop and another group below the loop. We call them "upper" and "lower" bays, respectively. The terms "above" and "below" can be defined arbitrarily but should be used consistently. We assign directions to the edges of a bay in the same manner.

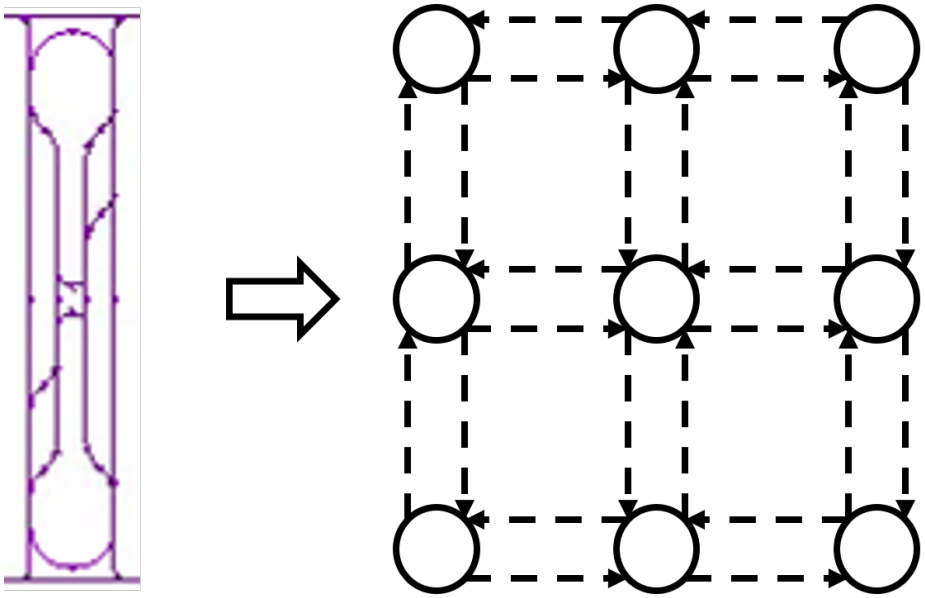


Figure 7: Simplification of bays in G ($BW = 3$, $BH = 3$)

The number of column nodes, (Column), depends on the number of bays, and the number of row nodes, (Row), relates to the number of center and outer loop lanes.

$$
\begin{aligned}
\text{(Column)} =& \text{(Outer loop)} + \text{(Horizontal bay nodes)} + \text{(Connection between two areas)} \\
=& 2 \times (NO + 1) + BW \times (NB/2) + BB \times (NB/2 - 1) + 2, \\
\text{(Row)} =& \text{(Outer loop)} + \text{(Vertical bay nodes)} + \text{(Center loop)} \\
=& 2 \times NO + 2 \times BH + 2 \times NC.
\end{aligned}
$$

We number the nodes from 1 to $|N|$. The topmost and leftmost node is node 1. Its right node and bottom node are 2 and 1 + (Column), respectively. We use this numbering for the constraints of a base track layout. If we assume that $BW = 3$, $BH = 3$, and $BB = 1$, then track layouts with 10 bays (5 upper and 5 lower bays), $NC = 1$ (one additional center loop lane), and $NO = 1$ (one additional outer loop lane) need $25 \times 10 = 250$ nodes. Figure 8 shows the grid graph.

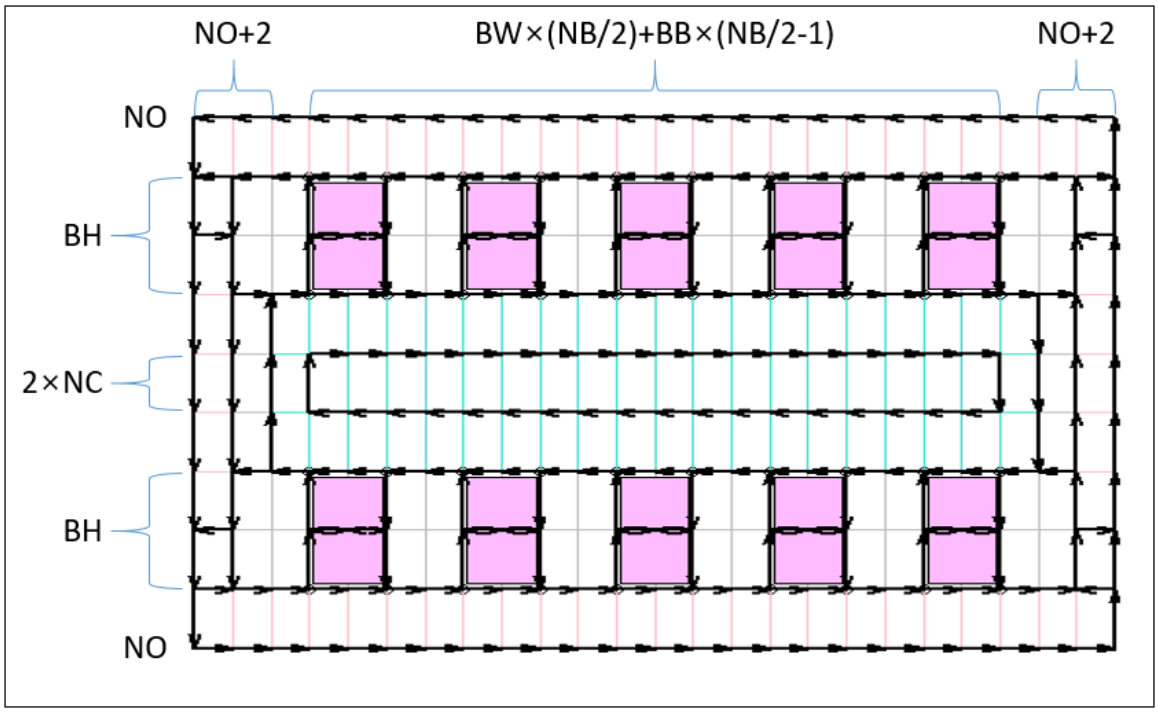


Figure 8: Grid graph when $NB = 10$, $DC = 0$, $DO = 0$, $BW = 3$, $BH = 3$, and $BB = 1$

3.2.3 Assumptions on commodities and alternative paths

We construct commodities reflecting the traffic based on transfer requests generated by the AMHS in a fab. We assume that transfer requests are generated based on the simulation model in [15]. Transfer requests depend on the locations of machines and the sequence of processing steps. Transfer requests can be prioritized, but the AMHS, in general, does not apply any special policy to the vehicles assigned to urgent transfer requests. Hence, the origin and destination of a transfer request matter to the assigned vehicle, and its frequency affects the traffic between two locations. We aim to incorporate the traffic from transfer requests in commodities, so commodities are defined by origin, destination, and frequency of requests. See Chapter 4 for the details.

In our formulation, we duplicate commodities to represent the two flow types (over the uncongested shortest path and over the alternative paths). We need to resolve two issues regarding the flow over alternative paths. Different commodities have different usages of their uncongested shortest paths. In addition, vehicles can take multiple alternative paths, and the number of used paths depends on various factors, such as a track layout, transfer request generation, etc. We provide the answers when we define the decision variables, objective function, and constraints.

3.3 Dynamic routing and alternative paths

A general multi-commodity NDP, which searches for a solution minimizing the shortest path of each commodity, is not suitable for our problem because dynamic routing makes vehicles take a variety of paths from one location to another depending on traffic conditions. The literature on vehicle navigation systems has studied how to provide alternative paths that satisfy user requirements. Abraham et al. [1] and other researchers ([10], [55], [36], [56], [76], [63]) proposed three conditions characterizing alternative paths:

Limited sharing The length of shared edges between the shortest and alternative paths does not exceed a certain fraction γ_1 of the length of the shortest path.

Local optimality Every subpath of an alternative path, the length of which is less than or equal to a certain fraction γ_2 of the shortest path, is optimal.

Uniformly bounded stretch For each subpath of an alternative path, let s and t be its origin and destination nodes, respectively. The subpath has cost bounded by a certain multiple γ_3 of the shortest path from s to t.

The existence of alternative paths depends on a graph and the three parameters, γ_1, γ_2, and γ_3. For example, a directed tree does not have any alternative path from the root node to any of its leaf nodes regardless of input parameters. Figure 9 shows a graph with no alternative path because of the limited sharing condition: there is no alternative path when $\gamma_1 < 10/14$. All of the paths from s to t have to use the edge of cost 10, but the shortest path has a cost of 14.

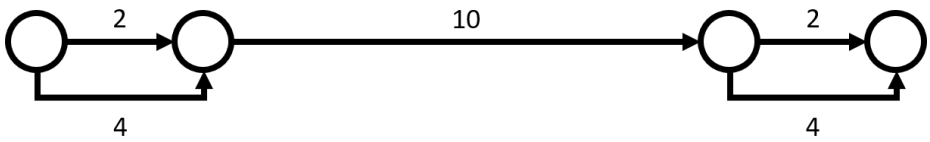


Figure 9: No alternative path when $\gamma_1 \leq 10/14$

Because finding proper values of the parameters is not trivial, we restrict the number of shared edges and minimize the cost of alternative paths. Specifically, we impose constraints on the portion of edges on the shortest path that are shared by alternative paths. As long as the paths satisfy the constraints, they are eligible alternative paths, but whether they are selected or not depends on their costs. Every feasible design is a bounded-degree graph because it has unidirectional lanes, two types of intersections, and no parallel lanes. After a vehicle passes an intersection, it cannot change its path until it approaches a diverging intersection. In addition, the

number of vehicles in front can increase or decrease at intersections because there is no parallel lane.

3.4 Formulation

3.4.1 Decision variables

Selecting a track layout that optimizes an objective function of the two types of flows yields the decision variables:

- $y \in \{0,1\}^{|E|}$ are the design variables: $y_{(i,j)} = 1$ if $(i,j) \in E$ is selected,
- $SP \in [0,1]^{|E|\times|K|}$ are the flow variables for the shortest paths of commodities, and
- $AP \in [0,1]^{|E|\times|K|}$ are the flow variables for alternative paths of commodities.

For each origin-destination pair, we have duplicated commodities. One is for SP, and the other is for AP. For any distinct commodities k_1 and k_2, we do not consider their interactions between SP^{k_1} and SP^{k_2} and between AP^{k_1} and AP^{k_2}. We only consider the relationship between SP^{k_1} and AP^{k_1} and between SP^{k_2} and AP^{k_2}.

To make SP^k and AP^k conform to our assumptions, we add two decision variables, π and s.

- $\pi \in \mathbb{R}_+^{|N|\times|K|}$ are the dual variables of the shortest path problem, which make SP^k the shortest path flow of commodity k.
- $s \in [0,1]^{|E|\times|K|}$ are the decision variables describing the relationship between two flow variables SP and AP.

3.4.2 Constraints

Constraints explain the relationship between two flow variables SP and AP and define a base track layout. The constraints for a base track layout select edges of E so that the solution design forms a spine layout. They are exclusive to designing a track

layout of a fab AMHS and may vary based on the application of interest. We simplify the constraints as $y \in \mathcal{Y}$. See Appendix for the details. Figure 10 is an example base track; the pink squares are bays, the black arrows are preselected edges, and the green arrows are candidate shortcut locations.

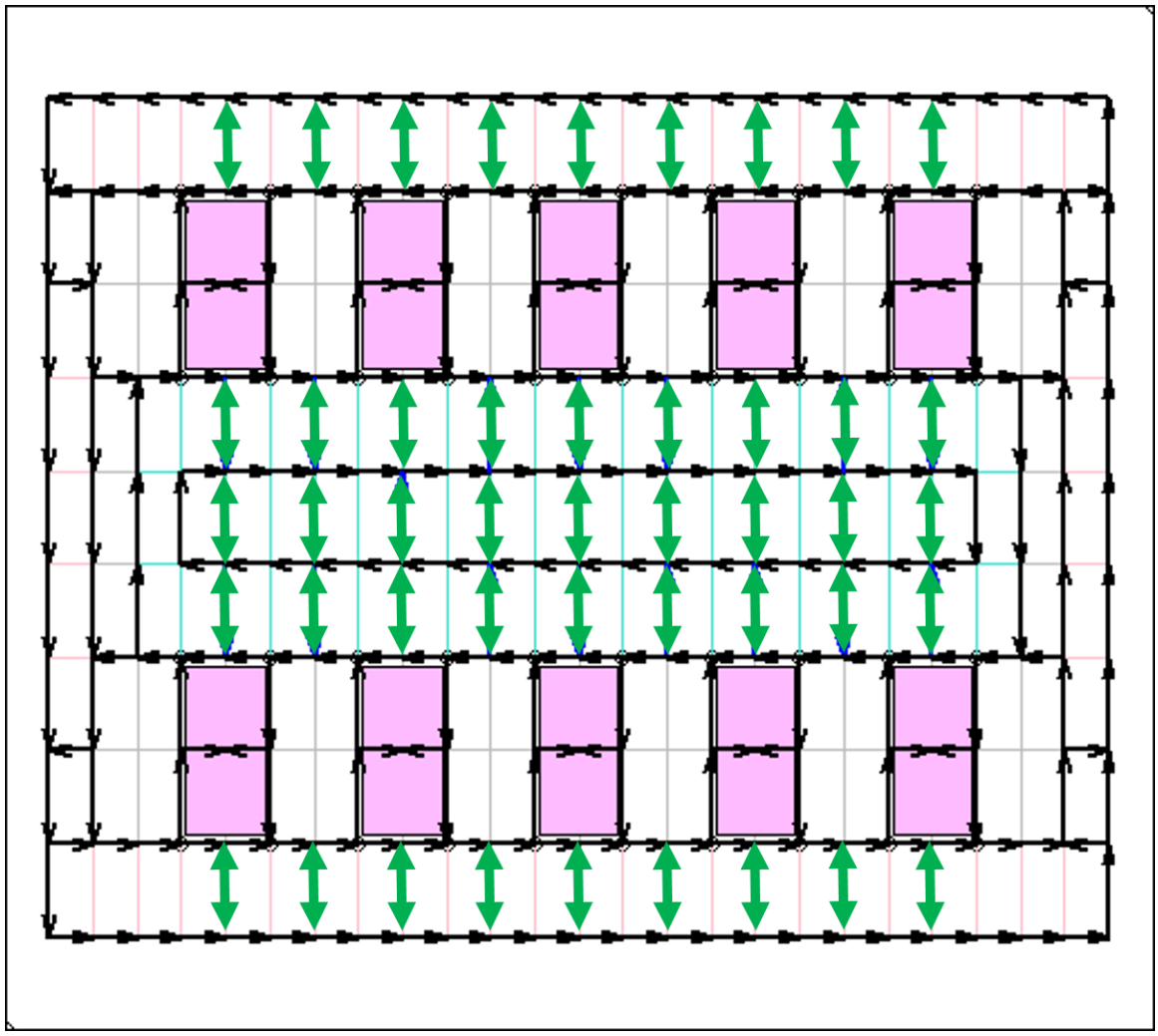


Figure 10: An example of a base track

Flow variables SP and AP satisfy the flow balance condition of each node.

$$\sum_{j:(i,j)\in E} AP^k_{(i,j)} - \sum_{j:(j,i)\in E} AP^k_{(j,i)} = \begin{cases} 1 & i = o_k, \\ -1 & i = d_k, \\ 0 & \text{otherwise}, \end{cases} \qquad \forall i \in N, \forall k \in K, \quad (1)$$

$$\sum_{j:(i,j)\in E} SP^k_{(i,j)} - \sum_{j:(j,i)\in E} SP^k_{(j,i)} = \begin{cases} 1 & i = o_k, \\ -1 & i = d_k, \\ 0 & \text{otherwise}, \end{cases} \qquad \forall i \in N, \forall k \in K, \quad (2)$$

$$AP^k_{(i,j)} \le y_{(i,j)},\ SP^k_{(i,j)} \le y_{(i,j)} \qquad \forall (i,j) \in E, \forall k \in K, \quad (3)$$

$$AP^k_{(i,j)} \in [0,1],\ SP^k_{(i,j)} \in [0,1] \qquad \forall (i,j) \in E, \forall k \in K. \quad (4)$$

For each commodity, no alternative path is shorter than the shortest path:

$$c^T AP^k \ge c^T SP^k. \quad (5)$$

SP^k should be the shortest path flow of commodity k. We employ the dual formulation of the shortest path problem to enforce the condition that SP^k is the shortest path:

$$\max_{\pi} \left\{ \pi_{d_k} - \pi_{o_k} \middle| \pi_j - \pi_i \le c_{(i,j)},\ \forall (i,j) \in E;\ \pi_i \ge 0, \forall i \in N \right\} \quad (6)$$

For each commodity $k \in K$, $\pi_{d_k} \le c^T SP^k$ because of weak duality, and the equality holds when $c^T SP^k$ equals to the shortest path cost. Accordingly, we introduce the following constraints:

$$\pi_{d_k} \ge c^T SP^k \qquad \forall k \in K, \quad (7)$$

$$\pi_{o_k} = 0 \qquad \forall k \in K, \quad (8)$$

$$\pi_j - \pi_i \le c_{(i,j)} + M(1 - y_{(i,j)}) \qquad \forall (i,j) \in E, \forall k \in K. \quad (9)$$

In (9), M is an arbitrary large number; it activates the constraint when $y_{(i,j)} = 1$, i.e., edge (i,j) is selected. Example 1 shows that without the constraints (7)-(9), SP^k is not guaranteed to be the shortest path.

Example 1. Table 1 shows the costs of four paths A, B, C, and D and the portion of shared edges for each pair of paths. For example, path B shares 80% of edges on path A, and path C shares 30% of edges on path B. We want to minimize

Table 1: Path selection example

	Cost	A	B	C	D
A	10	1.0	0.8	0.7	0.3
B	14		1.0	0.3	0.7
C	16			1.0	0.8
D	18				1.0

$0.3 \times SP + 0.7 \times AP$. By definition, $SP \leq AP$, and SP and AP must not share edges more than 50%. Thus, the feasible path pairs are (A, D) and (B, C). Because SP is the shortest path, the correct answer is $SP = A$ and $AP = D$. Without the constraints enforcing SP to be the shortest, $SP = B$ and $AP = C$ will be chosen because

$$0.3 \times 14 + 0.7 \times 16 < 0.3 \times 10 + 0.7 \times 18.$$

Proposition 1. *Every feasible solution (y, SP, AP, π) satisfies $SP^k_{(i,j)} = 1$ if and only if (i, j) belongs to the shortest path from o_k to d_k for each $k \in K$ and for each $e \in E$.*

Proof. Suppose that we have a subgraph defined by $y \in \mathcal{Y}$. Finding the shortest path from o_k to d_k is

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in E: y_{(i,j)}=1} c_{(i,j)} SP^k_{(i,j)} \\
\text{sub. to} \quad & \sum_{j:(i,j)\in E, y_{(i,j)}=1} SP^k_{(i,j)} - \sum_{j:(j,i)\in E, y_{(j,i)}=1} SP^k_{(j,i)} = \begin{cases} 1 & i = o_k, \\ -1 & i = d_k, \\ 0 & \text{otherwise}, \end{cases} \qquad \forall i \in N, \\
& SP^k_{(i,j)} \in [0, 1] \quad \forall (i, j) \in E : y_{(i,j)} = 1.
\end{aligned}$$

We call this problem $PSP^k(y)$, and then its dual $DSP^k(y)$ is

$$\begin{aligned}
\max \quad & \pi_{d_k} \\
\text{sub. to} \quad & \pi_j - \pi_i \leq c_{(i,j)} \qquad \forall (i, j) \in E : y_{(i,j)} = 1, \\
& \pi_{o_k} = 0.
\end{aligned}$$

Let $(\bar{y}, \bar{SP}, \bar{AP}, \bar{\pi})$ satisfy constraints (1)-(9). Then, $\bar{SP}^k$ satisfy the constraints of $PSP^k(\bar{y})$ because of (2), (3), and (4). Similarly, (7), (8), and (9) make $\bar{\pi}^k$ feasible for $DSP^k(\bar{y})$. Hence, it satisfies $c^T\bar{SP}^k \geq \bar{\pi}_{d_k}$ by weak duality, and (7) leads to $c^T\bar{SP}^k = \bar{\pi}_{d_k}$. Strong duality implies that $\bar{SP}^k$ represents the flow over the shortest path from o_k to d_k. □

We restrict the number of edges of the shortest path shared by alternative paths. The constraints for this condition involve the input parameter η and the decision variables s, SP, and AP. Because we allow AP to be fractional, s^k_e corresponds to the portion of the shortest path flow affected by alternative paths.

$$s^k_{(i,j)} \leq SP^k_{(i,j)} \qquad \forall (i,j) \in E, \forall k \in K, \tag{10}$$

$$s^k_{(i,j)} \leq AP^k_{(i,j)} \qquad \forall (i,j) \in E, \forall k \in K, \tag{11}$$

$$s^k_{(i,j)} \geq SP^k_{(i,j)} + AP^k_{(i,j)} - 1 \qquad \forall (i,j) \in E, \forall k \in K, \tag{12}$$

$$\sum_{(i,j)\in E} s^k_{(i,j)} \leq (1-\eta)1^T SP^k, \qquad \forall k \in K. \tag{13}$$

Constraints (10) and (11) imply $s^k_{(i,j)} = 0$ if edge (i,j) is not on the shortest path of commodity k, or if the edge is on the shortest path but $AP^k_{(i,j)} = 0$, i.e., no alternative path uses the edge. Otherwise, $s^k_{(i,j)} \geq AP^k_{(i,j)}$ because of the constraint (12). Then, constraint (13) imposes the upper bound on the sum of $s^k_{(i,j)}$ over every $(i,j) \in E$. Note that $1^T SP^k$ is the number of edges on the shortest path of commodity k.

3.4.3 Objective function

The objective function is to minimize the weighted sum of the cost functions of commodities; its weight is w_k, the flow intensity of commodity k. Let $\text{Cost}_k(y)$ be the cost function of commodity k. Then, the objective function is

$$\epsilon \sum_{(i,j)\in E} y_{(i,j)} + \sum_{k\in K} w_k \text{Cost}_k(y),$$

where ϵ is a small positive number that prevents unnecessary edges from being selected.

We define the cost function of a commodity to be the convex combination of two flow costs. $c^T SP^k$ is the cost of the shortest path of commodity k. It equals the average travel time of the shortest path if $c_{(i,j)}$ is the average travel time of edge (i,j). $c^T AP^k$ is the cost of alternative paths, which we explain now.

We allow AP_e^k to have a fractional value so that it allows for the relative usage of edge e by the alternative paths. Vehicles can take multiple alternative paths, and their usages can vary. Let $P_1^k, \ldots, P_{p_k}^k$ and $r_1, \ldots, r_{p_k}$ be alternative paths for commodity k and their usages. Different commodities may use different numbers of alternative paths. To represent this, AP has the following value:

$$AP_{(i,j)}^k = \frac{r_1 1_{\{(i,j)\in P_1\}}}{r_1 + \cdots + r_{p_k}} + \cdots + \frac{r_{p_k} 1_{\{(i,j)\in P_{p_k}\}}}{r_1 + \cdots + r_{p_k}}$$

Example 2 illustrates the value of alternative path flow variables.

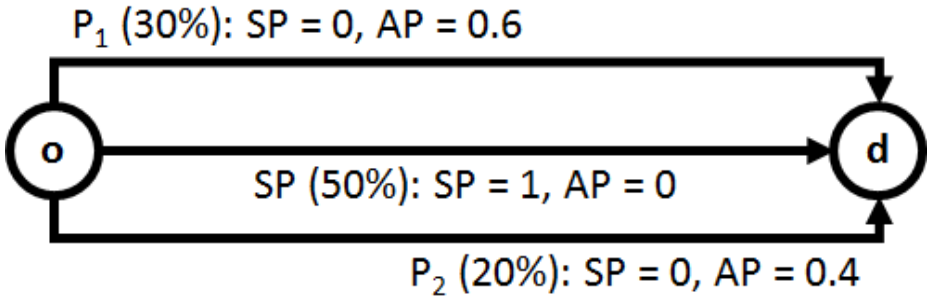


Figure 11: Multiple alternative paths and the value of AP

Example 2. Figure 11 shows that 50% of transfer requests from o to d use the shortest path (SP), and 30% and 20% of transfer requests use P_1 and P_2. The alternative path

flow variables are

$$AP_{(i,j)} = \begin{cases} 30\%/(30\% + 20\%) = 0.6 & \text{if } (i,j) \in P_1 \setminus P_2, \\ 20\%/(30\% + 20\%) = 0.4 & \text{if } (i,j) \in P_2 \setminus P_1, \\ 0.6 + 0.4 = 1 & \text{if } (i,j) \in P_1 \cap P_2, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, we define $c^T AP^k$ to be the weighted average of the cost of the alternative paths based on their usages. Nevertheless, it may not be possible to specify all of the alternative paths for each commodity if alternative paths share more edges with each other as sown in Example 3.

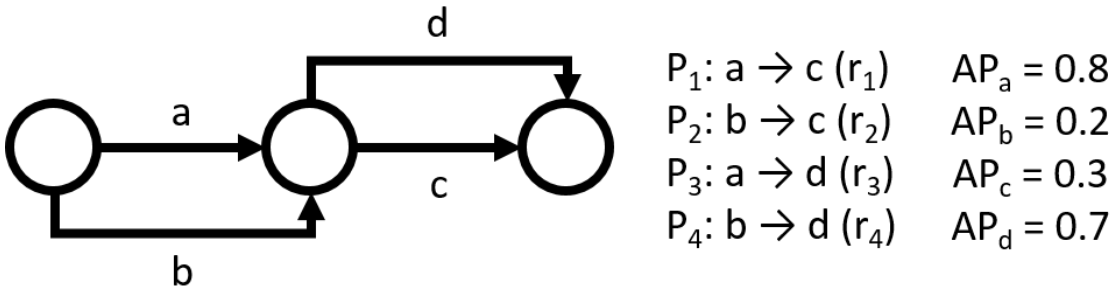


Figure 12: An example of multiple alternative paths

Example 3. Figure 12 shows the values of AP on four edges: $AP_a = 0.8$, $AP_b = 0.2$, $AP_c = 0.3$, and $AP_d = 0.7$. From these values, we may expect that four alternative paths P_1, P_2, P_3, and P_4 are used. Let r_1, r_2, r_3, and r_4 be the usage of the four alternative paths, respectively. Can we track the usage correctly? We have four equations:

$$\begin{aligned} a&: \quad r_1 + r_3 = 0.8 \\ b&: \quad r_2 + r_4 = 0.2 \\ c&: \quad r_1 + r_2 = 0.3 \\ d&: \quad r_3 + r_4 = 0.7 \end{aligned}$$

The equation system is not linearly independent, so it has multiple solutions. In other words, multiple combinations of alternative paths and their usages can satisfy the relationship.

We combine $c^T SP^k$ and $c^T AP^k$ to define $\text{Cost}_k(y)$, the cost of commodity k, using λ_k. It implies how often the traffic flow of commodity k deviates from its shortest path. Hence, we have

$$\text{Cost}_k(y) := \lambda_k c^T AP^k + (1 - \lambda_k) c^T SP^k,$$

where SP^k and AP^k depend on y. Finally, the overall objective function is

$$\epsilon \sum_{(i,j)\in E} y_{(i,j)} + \sum_{k\in K} w_k \left(\lambda_k c^T AP^k + (1 - \lambda_k) c^T SP^k\right).$$

The full formulation is as follows. For notational simplicity, we call this problem NDPA, or NDPA(η) if the value of η matters.

$$\begin{aligned}
\min_{y,AP,SP,\pi,s} \quad & \epsilon \sum_{(i,j)\in E} y_{(i,j)} + \sum_{k\in K} w_k \left(\lambda_k c^T AP^k + (1 - \lambda_k) c^T SP^k\right) \\
\text{sub. to} \quad & y \in \mathcal{Y}, \\
& AP^k_{(i,j)} \le y_{(i,j)},\ SP^k_{(i,j)} \le y_{(i,j)} && \forall (i,j) \in E, \forall k \in K, \\
& \sum_{j:(i,j)\in E} AP^k_{(i,j)} - \sum_{j:(j,i)\in E} AP^k_{(j,i)} = \begin{cases} 1 & i = o_k, \\ -1 & i = d_k, \\ 0 & \text{otherwise}, \end{cases} && \forall i \in N, \forall k \in K, \\
& \sum_{j:(i,j)\in E} SP^k_{(i,j)} - \sum_{j:(j,i)\in E} SP^k_{(j,i)} = \begin{cases} 1 & i = o_k, \\ -1 & i = d_k, \\ 0 & \text{otherwise}, \end{cases} && \forall i \in N, \forall k \in K, \\
& c^T AP^k \ge c^T SP^k && \forall k \in K, \\
& \sum_{(i,j)\in E} s^k_{(i,j)} \le (1 - \eta) 1^T SP^k && \forall k \in K, \\
& s^k_{(i,j)} \ge SP^k_{(i,j)} + AP^k_{(i,j)} - 1 && \forall (i,j) \in E, \forall k \in K,
\end{aligned}$$

$$
\begin{aligned}
& s^k_{(i,j)} \le SP^k_{(i,j)} && \forall (i,j) \in E, \forall k \in K, \\
& s^k_{(i,j)} \le AP^k_{(i,j)} && \forall (i,j) \in E, \forall k \in K, \\
& \pi_{d_k} = c^T SP^k && \forall k \in K, \\
& \pi_{o_k} = 0 && \forall k \in K, \\
& \pi_j - \pi_i \le c_{(i,j)} + M(1 - y_{(i,j)}) && \forall (i,j) \in E, \forall k \in K, \\
& AP^k_{(i,j)} \in [0,1],\ SP^k_{(i,j)} \in [0,1] && \forall (i,j) \in E, \forall k \in K, \\
& \pi_i \ge 0 && \forall i \in N, \forall k \in K, \\
& s^k_{(i,j)} \in [0,1] && \forall (i,j) \in E, \forall k \in K.
\end{aligned}
$$

Note that the optimal design of NDPA is also optimal for a classic multi-commodity network design problem when $\lambda = 0$. It also holds when $\eta = 0$ and the shortest path of each commodity is unique.

An optimal solution of NDPA may assume two or more alternative paths for a commodity. Given $y \in \mathcal{Y}$, we can find the shortest path p^k_0 for commodity k. In addition, let $\{p^k_0, p^k_1, \ldots, p^k_{r_k}\}$ denote the set of all the paths from o_k to d_k. Two properties of path p^k_i are its cost (c_i) and the number of edges of p^k_0 which p^k_i shares (s_i). We assume that s_0 is the number of edges on p_0. Let $x^k_0, x^k_1, \ldots, x^k_{r_k}$ be decision variables of the portion amount of flow on the path. We do not consider the interaction of multiple commodities, and we force the fixed amount of flow along the shortest path. Hence, our problem is equivalent to the following linear program:

$$
\sum_{k \in K} \left((1 - \lambda_k) c^k_0 + \lambda_k c^k_{\text{alt}}, \right).
$$

where

$$
c^k_{\text{alt}} = \min_p \left\{ \sum_{i=1}^{r_k} c^k_i p^k_i \;\middle|\; \sum_{i=1}^{r_k} s^k_i p^k_i \le (1-\eta) s^k_0,\ \sum_{i=1}^{r_k} p^k_i = 1,\ p^k_i \ge 0\ \forall i = 1, \ldots, r_k \right\}.
$$

While any extreme points have at most two nonzero entries, multiple extreme points can be optimal, and thus any of their convex combinations is also optimal. Hence, an optimal solution can have more than two nonzero entries.

NDPA provides a track design based on static input data, but track layouts should consider dynamic changes in traffic conditions. To make it more realistic, we integrate optimization and simulation in Chapter 4. We fix the value of $\eta \in [0, 1]$ and obtain a track layout from the iterations between the optimization and simulation. The computational results in Chapter 5 show that selecting a proper value of η makes significant improvements in routing performance. We call such η "the best η."

3.4.4 Additional approaches

We present two additional scenario-based approaches to find a track layout of a fab AMHS. Both resemble network interdiction problems, i.e., edge failures affect edge selections. Let S be the set of scenarios. In scenario s, we assume that e_s suffers from heavy congestion and that no vehicles can traverse it. Since we do not incorporate alternative path flows, we have single type of flow variables $x \in [0, 1]^{|E|\times|K|\times|S|}$; x_e^k is the flow of commodity k over edge e in scenario s. We plan to search for a solution robust to every scenario.

The two approaches have different objective functions:

- The sum of the minimum total travel time over scenarios:

$$\sum_{s\in S}\left(\min_{x}\left\{\sum_{k\in K} w_k \sum_{e\in E} c_e x_e^k(s) \;\middle|\; x(s)\in\mathcal{X}(y),\ \forall s\in S;\ \sum_{k\in K} x_{e_s}^k(s)=0,\ \forall s\in S\right\}\right) \tag{14}$$

- The maximum of the minimum total travel time over scenarios:

$$\max_{s\in S}\left\{\min_{x}\left\{\sum_{k\in K} w_k \sum_{e\in E} c_e x_e^k(s) \;\middle|\; x(s)\in\mathcal{X}(y),\ \forall s\in S;\ \sum_{k\in K} x_{e_s}^k(s)=0,\ \forall s\in S\right\}\right\} \tag{15}$$

We linearize them as:

$$(14): \qquad \min_{y,x} \quad \epsilon\sum_{e\in E}+\sum_{s\in S} z_s$$

$$\text{sub. to} \quad y\in\mathcal{Y},$$

$$
\begin{aligned}
& x(s) \in \mathcal{X}(y) && \forall s \in S, \\
& \sum_{k \in K} x^k_{e_s}(s) = 0 && \forall s \in S, \\
& z_s \geq \sum_{k \in K} w_k \sum_{e \in E} c_e x^k_e(s) && \forall s \in S.
\end{aligned}
$$

$$
\begin{aligned}
(15): \quad & \min_{y,x} \quad \epsilon \sum_{e \in E} + z \\
& \text{sub. to} \quad y \in \mathcal{Y}, \\
& x(s) \in \mathcal{X}(y) && \forall s \in S, \\
& \sum_{k \in K} x^k_{e_s}(s) = 0 && \forall s \in S, \\
& z \geq \sum_{k \in K} w_k \sum_{e \in E} c_e x^k_e(s).
\end{aligned}
$$

In Chapter 5, we present the computational results of the optimization problems (14) and (15). We plug them as well as NDPA into the integrated framework proposed in Chapter 4. We examine a variety of test cases, and in some of them, the design from (14) or (15) shows routing performance similar to the layout from NDPA with the best η. However, none of them is consistently better than NDPA with $\eta = 0$. We suspect that the number of scenarios or completely blocking congested edges should be investigated in detail.

3.5 Conclusions

We propose a network design problem that incorporates alternative paths. We define alternative paths for commodities to represent the traffic under dynamic routing. Dynamic routing guides vehicles to less congested locations and changes their paths if necessary. Multiple alternative paths can be used based on traffic conditions, which is represented as multiple fractional flows, and the relationship between the shortest and alternative paths depends on rerouting frequencies of a commodity. We formulate this problem as a variation of multi-commodity network design problems.

Solution designs are feasible track layouts of a fab AMHS, and we expect them to increase the efficiency of material handling. However, traffic conditions of an AMHS are changing continually while a track layout is static. To fill the gap, we integrate optimization and simulation in Chapter 4. In order to find the best design, we construct problem instances by changing η. For each instance, we repeat the iteration of optimization and simulation until we obtain a design. Computational results are presented in Chapter 5. We observe that a specific value of η provides a design that outperforms others from different values of η.

CHAPTER IV

INTEGRATION OF OPTIMIZATION AND SIMULATION FOR AN AMHS TRACK DESIGN

4.1 Introduction

Optimization presents an optimal solution with respect to an objective function and constraints, and simulation represents the realistic behaviors of a target system. Figure 13 outlines our combined approach. The optimization problem NDPA in Chapter 3 incorporates alternative paths and rerouting frequencies to reflect the traffic under dynamic routing. Its input data, especially, the cost vector c and rerouting frequencies λ_k, $k \in K$, depend on the actual vehicle movements. We employ the simulation model presented by Bartlett [15] to validate the design from NDPA and provide its input data.

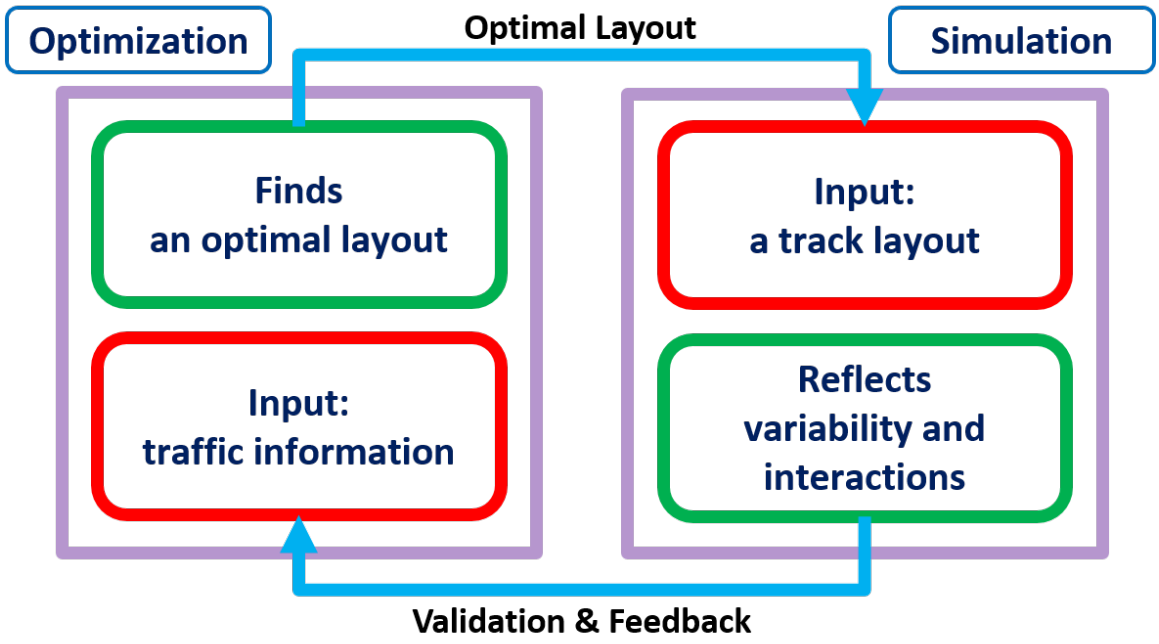


Figure 13: Combining optimization and simulation

Combining the optimization and simulation poses a few issues. First, track segments in simulation models and edges in NDPA do not always match. For example, it frequently happens that one track segment corresponds to multiple concatenated edges. Second, we need to define the original path for a transfer. Then, we can monitor rerouting operations. Because traffic conditions are changing continuously, the system can dispatch vehicles to different paths. Third, we need to select the best design during the optimization-simulation iterations. Hence, knowing when to terminate the iteration and what to choose are crucial decisions.

Section 4.2 describes the simulation model and the model generator. Section 4.3 presents the method to generate the input data of NDPA from simulation results. Section 4.4 presents a heuristic that combines optimization and simulation to obtain a solution design.

4.2 Simulation

4.2.1 Model description

We employ the simulation model presented in [16] and [15] to measure routing performance and provide the input data of NDPA in Chapter 3. The simulation model represents the vehicle operations of a fab AMHS. Below, we describe the transfer requests, vehicles, and dynamic routing method embedded in our simulation.

To generate realistic transfer requests, a method based on production information was proposed in [16] and [15]. Both studies constructed a Markov chain using a product mix, the sequence of processing steps of each product, and the assignment of processing steps to bays. For each product, a transition matrix on the state space consisting of processing steps was built and its stationary distribution was calculated. The state space consisted of processing steps, and the stationary distribution described the probability that the origin of a transfer request was a specific processing step. Given the origin of a request, the transition matrix stated the probability

that its destination was a specific step. The stopping locations (for unloading and loading) for the assigned vehicle were selected after the two processing steps for the origin and the destination were specified. After the warm-up period, the origin was selected based on either the stationary distribution or the number of net deliveries in each bay.

In our simulation, we assume that the origin of a transfer request is selected based on the stationary distribution at all times because our problem does not monitor work-in-process levels of the bays. We assume ten seconds for unloading and unloading. We impose one meter of distance between two vehicles for safety. Idle vehicles remain at their locations unless an assigned vehicle is coming close. Then, idle vehicles proceed to the closest diverging intersection and select the branch track that the assigned vehicle will not select. All vehicles move at different velocities on straight and curved lanes, and may have to decelerate or accelerate.

We use the dynamic routing method, "learn-and-adapt," which reflects up-to-date traffic conditions and directs vehicles at diverging intersections. The method estimates the travel time of each track segment using the exponential weighted moving average of historical data. Using the travel time data, the method periodically calculates one-to-all shortest paths from diverging intersections and stores the path information in lookup tables. In simulation, whenever a vehicle approaches a diverging intersection, the AMHS uses the lookup table to "tell" the vehicle its next location.

4.2.2 Simulation generator

In our dissertation, we compare different track designs using simulation. Because the simulation model requires a track layout as input data, we need to build simulation models with different setups. If we generate a simulation model swiftly, then it helps us select a good design faster. Bartlett [15] presented a simulation generator in her

dissertation, which fulfils this demand. We describe selected features of the simulation generator, which are related to our research.

The generator builds a simulation model based on user requirements in less than a minute. It requires layout input and operational input data to specify the track layout and generate transfer requests. Layout input data consist of:

- Number of bays and their structure.
- Number of center and outer loop lanes and their directions.
- Locations and directions of shortcuts in the center and outer loops.
- Connection types between the center and outer loops.
- Number of vehicles and their speeds, acceleration, and deceleration on straight and curved lanes.

Operational input data include:

- Bay-process assignment.
- Production information (product mix, transition matrix, stationary distribution).
- Routing methods (static, semi-dynamic, dynamic).
- Update frequency.
- Parameters for controlling vehicle breakdowns.

Users can make additional modifications. For example, track segments can be added or removed manually. Two sides of a bay may be assigned to different processing steps. In addition, users can change the size of a bay.

4.3 Feedback from simulation to optimization

After running the simulation, we analyze the results and calculate c and λ, which are the input data of NDPA. Then, we keep re-running the optimization and simulation until the termination criterion is met.

4.3.1 Commodities

Commodities should reflect the traffic based on transfer requests generated by the AMHS in a fab. We use the Markov chain embedded in the simulation model to determine the origin, destination, and demand of the commodities. As we investigate a shortcut placement problem for the center and outer loops, we define the commodities based on inter-bay transfer requests. Vehicle movements inside one bay can affect the traffic conditions in the center loop and the other bays, but we ignore them. We aggregate stopping locations according to their bays to reduce the problem size. Bays have more than 50 stopping locations, so more than 200 thousand origin-destination pairs are candidates for transfer requests. Hence, we define each commodity using its origin bay, destination bay, and demand. The origin and destination of a bay use the same node, and we assume they are the aggregation of all possible origin and destination locations in the bay. Then, we allow the terms o_k and d_k to denote the origin bay and the destination bay of commodity k.

The demand of a commodity derives from the frequency of transfer requests. We assume that pairs of bays with the same processing step pair have the same frequency of transfer requests. The frequency of a transfer request depends on a stationary distribution and a transition matrix. Let b_o and b_d be the origin and destination bays. We assign processing steps s_o and s_d to them, respectively, for which the fab has n_{s_o} and n_{s_d} machines. If we manufacture products $1, \ldots, r$ according to the product mix $(m_1, \ldots, m_r)$, then the method defines transition matrices $P^1, \ldots, P^r$ and stationary distributions $\pi^1, \ldots, \pi^r$. Then, the demand of the commodity from b_o

to b_d is

$$w_{b_o,b_d} = m_1 \frac{\pi^1_{s_o}}{n_{s_o}} \frac{p^1_{s_o,s_d}}{n_{s_d}} + \cdots + m_r \frac{\pi^r_{s_o}}{n_{s_o}} \frac{p^r_{s_o,s_d}}{n_{s_d}}. \tag{16}$$

In (16), we sum the multiplication of:

- The portion of product i in the product mix,
- The probability that the transfer request is for product i and has its origin at bay b_{s_o}, and
- The probability that the transfer request is for product i and has its destination at bay b_{s_d}, given that its origin is bay b_{s_o}.

In other words, each term implies for the probability that the transfer request is for product i and has its origin at bay b_{s_o} and its destination at bay b_{s_d}.

4.3.2 Average travel time

We collect the average travel time of each track segment in the simulation and convert it to one or more edge costs. When an edge in the optimization matches a base track segment, such as the one connecting the center and outer loops, then the edge cost is set to be the average travel time of the track segment. Our simulation, however, provides no information about unselected shortcuts. It also happens that two or more edges in the optimization match a base track segment between two adjacent intersections. Hence, we need to resolve the following issues:

- All track segments including shortcuts are unidirectional,
- Multiple edges are concatenated without an intersection for a long track segment, and
- The optimal track layout does not place shortcuts on all candidate shortcut locations.

The issues raise the following questions for which we provide the answers:

- **What is the edge cost if its direction is reversed?** We assign the same cost to edges with opposite directions. Reversed edges have the same shape and uncongested travel time, even though simulated travel time can differ. Moreover, it is impractical to build an additional simulation model whenever we flip a shortcut edge.

- **How should we allocate the average travel time of a single track segment to multiple edges?** When multiple edges are connected without an intersection, the simulation provides information about the long track segment. We assume that average traffic conditions are homogeneous on the long track segment, which is reasonable given that all track segments are unidirectional. Then, the portion of the average travel time of each edge is proportional to its length, or equivalently, its uncongested travel time. Suppose that $e_1, \ldots, e_s$ are connected with no intersection and form track segment r_{long}. Let $t_1, \ldots, t_s$ and $c_1, \ldots, c_s$ be the uncongested travel time of each edge and its cost in optimization, respectively. If it takes $\bar{t}_{long}$ to traverse r_{long} on average, then for every $i = 1, \ldots, s$, we have
$$c_i = \frac{t_i}{\sum_{i=1}^{s} t_i} \bar{t}_{long}.$$

- **What is the cost of unselected shortcut edges?** We assign uncongested travel time to unselected shortcut edges. In most cases, the uncongested travel time of an edge is smaller than the simulated average. Thus, it motivates the problem to choose unselected edges, and test more designs.

We only consider the average travel time of each track segment because it is difficult to capture its variability. For example, let two track segments r_a and r_b have the same uncongested travel time and the same coefficient of variation of travel time. We assume that r_a has suffered from chronic but mild congestion and that r_b was

mostly uncongested but affected by serious congestion during a short time period. No single summary statistic can accurately represent this difference. Alternatively, we can capture it by constructing a complicated time-series, but it is difficult to incorporate it in network design problems.

4.3.3 Rerouting frequency

Rerouting is the most important characteristic of dynamic routing, so we we define it formally and calculate its frequency. As mentioned, the "learn-and-adapt method" in [15] does not specify a complete path from origin to destination, but it does maintain the repeatedly updated lookup tables used by the AMHS to guide vehicles at diverging intersections, given a destination.

To define rerouting, we first define the original path for an origin-destination pair. The uncongested shortest path is a candidate, but it does not reflect actual traffic conditions. It underestimates the minimum travel time from the origin to the destination. In addition, uncongested travel time of an edge can vary based on the path to which the edge belongs to. Vehicles move at different velocities on straight and curved lanes, and intersections have at least one curved lane. Without congestion, vehicles are moving at different speeds on the same track segment based on neighbor intersections. Figure 14 illustrates four examples. Vehicles move from o to d, but their velocities on (a, b), v_2, v_3, and v_4, depend on the length of (a, b) and the vehicles' locations. Thus, it is difficult to define the uncongested travel time of edges, and also paths.

Instead, we define the original path dynamically. When a vehicle starts to move, we find the shortest path at the moment and designate it as the original path. In our simulation, the information in the lookup tables indicates the shortest path between two locations. Let $TR(o, d, t_0)$ be a transfer request from o to d generated at time t_0. Let n_o and n_d be the first intersection after o and the intersection closest to d. The

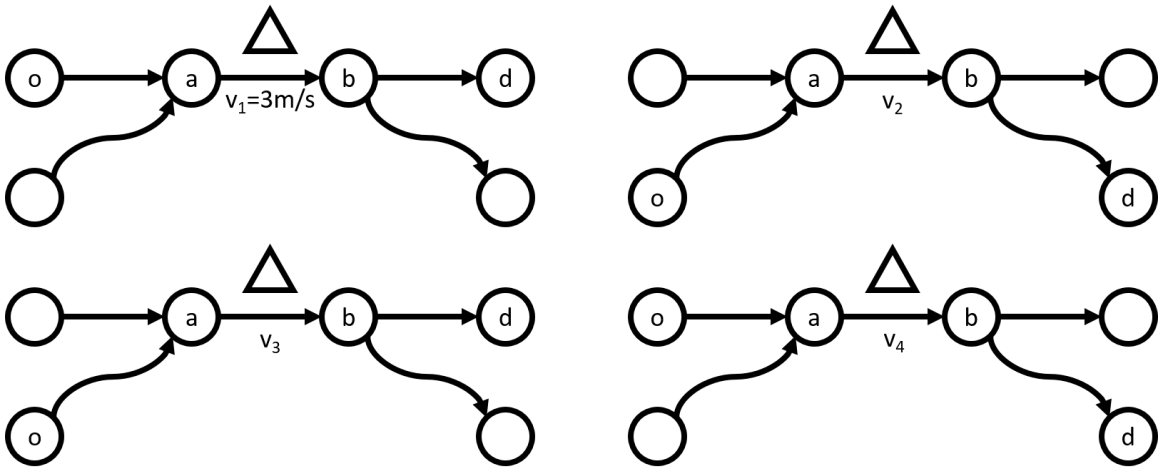


Figure 14: Different speeds at the same location

assigned vehicle finishes unloading at o at time $t \geq t_0$. We assume that the lookup tables are updated at t^- and t^+ satisfying $t^- \leq t < t^+$ and that no update is done during (t^-, t) or (t, t^+). Let $SP(o, d, t^-)$ be the shortest path from o to d based on the traffic conditions at t^-. Let n be a diverging intersection on $SP(o, d, t^-)$ and have two branch nodes n_a and n_b. We define that a vehicle is rerouted at n if it goes to n_b while $n_a \in SP(o, d, t^-)$.

As input data of the optimization, rerouting frequencies need to be consistent with the following two assumptions. First, we ignore idle vehicles because we aim to minimize the average travel time of FOUP transfers. Idle vehicles remain at the same locations unless assigned vehicles approach them. While yielding a path to assigned vehicles, idle vehicles do not have a specific destination. Second, we do not count additional rerouting operations after a vehicle deviates from its original path. Since traffic conditions are changing dynamically, rerouted vehicles may return to their original paths or execute subsequent rerouting operations. In Section 3.2, we duplicate each commodity so that we distinguish the traffic flow over the shortest path from others. One commodity represents the shortest path flow. We allow the other commodity, which shows the flow over alternative paths, to have fractional values on edges so that it represents the usage of multiple alternative paths.

Algorithm 1 calculates the rerouting frequencies based on estimated edge travel time and vehicle location data from the simulation. We aggregate the transfer requests according to their origin and destination bays according to our definitions of commodities in NDPA.

Algorithm 1 Calculation of rerouting frequencies

for all transfer request (o_k, d_k, t_k) **do**
  $o \leftarrow \text{Bay}(o_k)$; $d \leftarrow \text{Bay}(d_k)$
  $Total(o, d) \leftarrow Total(o, d) + 1$
  Find t_u and t_l when the unloading and loading start, respectively.
  Find $t^- \leq t_u$ such that the lookup tables are updated at t^-.
  Evaluate $SP(o, d, t^-)$.
  Find the assigned vehicle v.
  Load the records of v from $t_u + 10$ to t_l.
  Let $\bar{SP}(o, d)$ be the path of v.
  if $\bar{SP}(o, d) \neq SP(o, d, t^-)$ **then**
    $Rerouted(o, d) = Rerouted(o, d) + 1$.
  end if
end for
for all origin-destination pair (o, d) **do**
  $ReroutingFreq(o, d) \leftarrow Rerouted(o, d)/Total(o, d)$
end for

Finally, we define the weighted sum of rerouting frequencies according to the demand of the corresponding commodity:

$$ReroutingFreq = \sum_{k \in K} w_{(o_k, d_k)} ReroutingFreq(o_k, d_k),$$

which is reported in Chapter 5.

4.4 Integration of optimization and simulation

For a realistic problem, we combine optimization and simulation to design a track layout of a fab AMHS. We use the track layout provided by the optimization problem in Chapter 3 to build a simulation model using the simulation generator.

Simulation-optimization approaches (SO) could be a way to combine optimization and simulation with the consideration of budgetary constraints. However, it is difficult

to apply SO approaches to our problem, which has a discrete solution space because it determines shortcut placements. Most SO approaches for a discrete solution space, e.g., ranking and selection, random selection, stochastic ruler, and adaptive hyperbox, need to compare two neighbor solutions in order to decide the direction to explore. For instance, a small layout with 10 bays has 45 candidate shortcut locations, and each feasible design has more than 45 neighbor solutions. Moreover, the problem size is exponential with respect to the fab size. If we ignore node-degree constraints, then the number of feasible designs is $3^{38} \approx 10^{18}$.

In our study, we combine optimization and simulation by iteratively using them. As mentioned in [37], we need to determine two issues: the information that optimization and simulation provide to each other and when we terminate the iteration. Our approach has the following characteristics:

- From optimization to simulation: a track layout
- From simulation to optimization: c (the edge cost vector) and λ (the rerouting frequency vector)
- When to terminate the iteration: when a track layout has been already simulated

Using the simulation results, we calculate c and λ based on the approaches in Section 4.3. If the same track layout appears more than once, we terminate the iteration and select the layout that attains the smallest average travel time. Note that most of the relevant studies we cite in Chapter 2 updated their constraints using the simulation results while we update the objective function. It is because NDPA always generates a track layout that is feasible with respect to vehicle routing constraints.

To describe the procedure, Algorithm 2 describes the procedure, where we use two terms:

- NDPA(η, c, λ): the optimal solution design of NDPA with the uniqueness level η, the edge cost vector c, and the rerouting frequency vector λ
- Simulation(d): the cost and rerouting frequency vectors obtained from the simulation results of design d

Algorithm 2 Iterative approach between optimization and simulation

$D = \emptyset$; $c^0 \leftarrow$ uncongested edge costs; $\lambda^0 \leftarrow 0$
Design(0) $\leftarrow$ NDPA(η, c^0, λ^0)
$(c^1, \lambda^1) \leftarrow$ Simulation(Design(0))
$j \leftarrow 1$
repeat
    Design(j) $\leftarrow$ NDPA(η, c^j, λ^j)
    $(c^{j+1}, \lambda^{j+1}) \leftarrow$ Simulation(Design(j))
    $j \leftarrow j + 1$
    $D \leftarrow D \cup \{\text{Design}(j)\}$
until Design(j) $\in D$
return $d \in D$ with the least the average time in system

Before the iteration starts, we set $\lambda = 0$ and solve the problem using uncongested edge costs to obtain c^1 and λ^1. Once the iteration starts, the simulation at the previous stage provides c and λ, so we simplify the notation to NDPA(η).

Although the number of feasible solutions is finite, our approach does not guarantee that we obtain an outcome swiftly. However, it shows satisfactory results in the numerical examples given in Chapter 5. Figures 15 and 16 present the numbers of iterations of 10-bay and 20-bay problem instances. The horizontal axis is the number of iterations, and the vertical axis is the number of problem instances that terminate after the number of iterations specified by the horizontal axis. The total time to terminate the iteration depends on the fidelity of simulation and the computation time of optimization. The former does not change significantly during the iteration while the latter may vary based on the problem structure. However, all of the problem instances in our computational results are terminated in less than 50 hours, which could be shortened by any algorithm specialized to NDPAs.

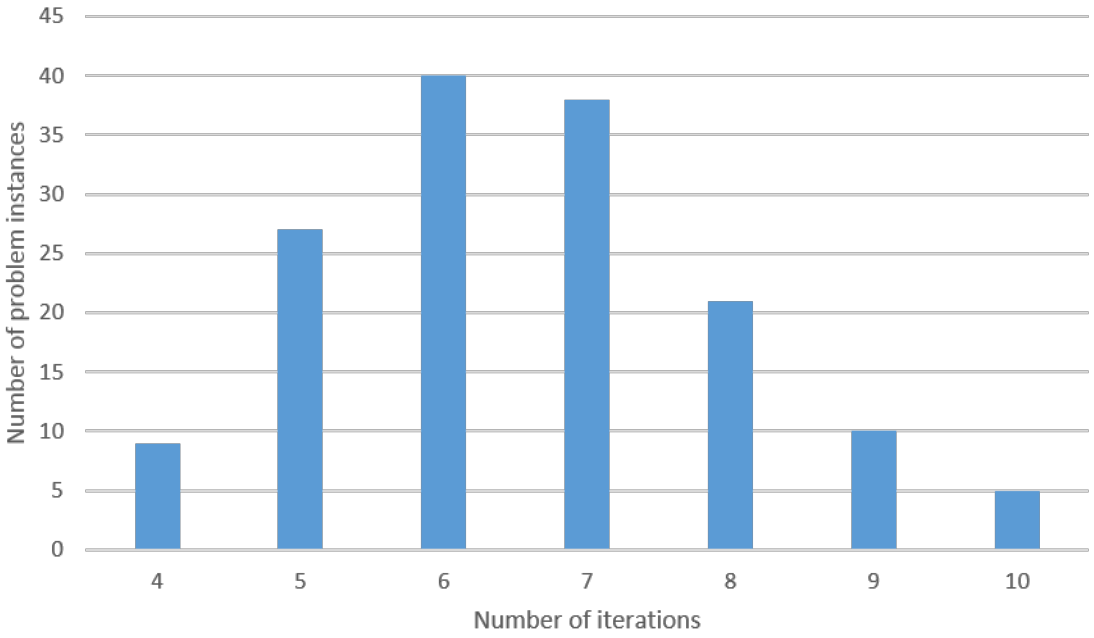


Figure 15: Number of iterations (10-bay)

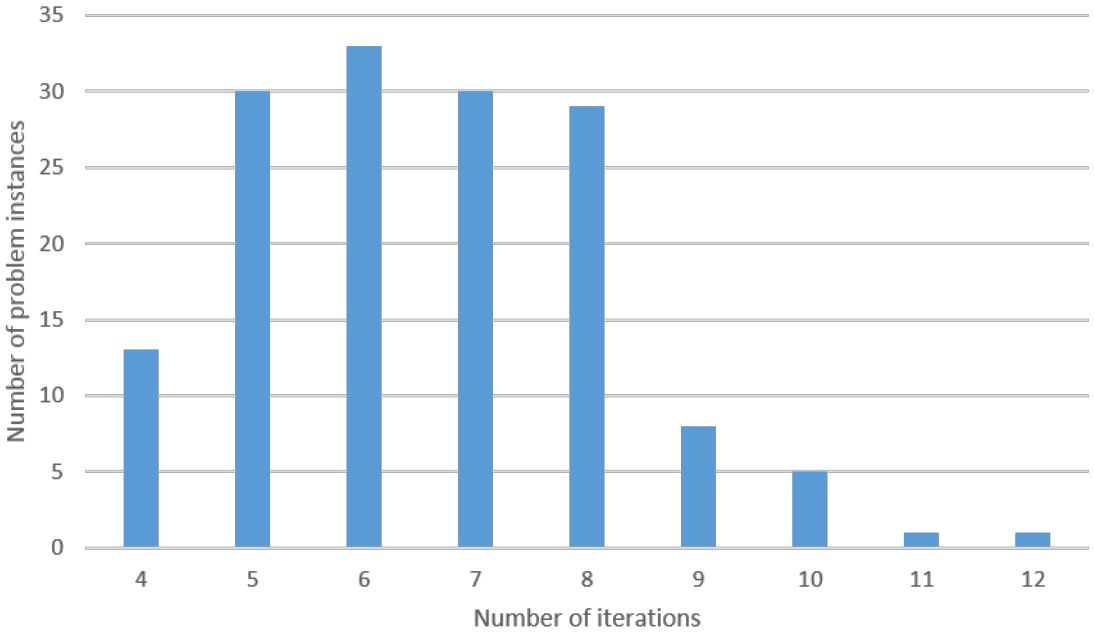


Figure 16: Number of iterations (20-bay)

CHAPTER V

COMPUTATIONAL RESULTS

This chapter presents the computational results of the shortcut placement problems, which are formulated as NDPAs in Chapter 3. We apply the optimization-simulation integration in Chapter 4 to obtain high quality designs by selecting the best track layout among those produced by the optimization and evaluated by the simulation.

We begin by fixing the number of bays, a bay-process assignment, and the initial workload level. Then, among the input data of the optimization, we control the uniqueness of alternative paths by adjusting η. Recall that we have two flow variables SP^k, the flow over the shortest path, and AP^k, the flow over alternative paths, for each commodity, and that η is the portion of the edges of SP^k shared by AP^k. Although we are unable to prove optimality, we did observe improved routing performance under certain values of η.

Section 5.1 describes the specifications of problem instances. Section 5.2 gives the base case results of the 10-bay and 20-bay instances. Section 5.3 reports two types of sensitivity analyses, investigates whether the designs are suitable for other routing schemes, and how the increased workload changes routing performance.

5.1 Problem specifications

The simulation models in this chapter share the following settings:

- Number of bays: 10 or 20
- Bay structure: 4-lane structure
- Bay with two processing steps: disabled

- Number and direction of center loop lanes: 4, parallel
- Number and direction of outer loop lanes: 2, parallel
- Connection type between the center and outer loops: wide-connect
- Product types: A, B (different predefined sequences of processing steps)
- Product mix: 0.5 and 0.5 for A and B
- Vehicle velocities on straight and curved lanes: 3 m/s, 1 m/s
- Vehicle acceleration and deceleration: 2 m/s^2, 3 m/s^2
- Time for loading and unloading: 10 s
- Number of vehicles: 200 for 10-bay instances and 250 for 20-bay instances
- Vehicle breakdowns: disabled
- Total run length and warm-up period: 13,200 s, 1,200 s
- Number of replications: 20

We make ten random bay-process assignments for 10-bay and 20-bay problems, respectively. In practice, machine locations and a track layout are decided concurrently. However, we focus on a shortcut placement problem and assume that the equipment layout is fixed. Hence, each bay-process assignment forms a separate problem instance. Tables 2 and 3 list the bay-process assignments we test. Transfer requests have their origin and destination among eight processing steps: cleaning (CLN, 1), diffusion (DIFF, 2), photolithography (PHT, 3), etching (ETCH, 4), ion implantation (IMP, 5), chemical vapor deposition (CVD, 6), metalization (MTL, 7), chemical-mechanical polishing (CMP, 8).

For the optimization problem, we assume that $BW = 3$, $BH = 3$, and $BB = 1$. We set the numbers of candidate locations between bay exit and entrance nodes

Table 2: Bay-process assignments for 10-bay layouts

	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
Bay01	1	6	8	8	4	4	2	5	7	3
Bay02	2	4	3	1	6	7	8	4	1	4
Bay03	3	7	1	1	2	5	4	1	2	1
Bay04	4	5	7	7	5	1	6	2	5	5
Bay05	5	1	6	4	8	8	5	8	1	1
Bay06	6	1	4	2	1	1	1	1	6	7
Bay07	7	3	4	5	1	6	4	7	4	4
Bay08	8	2	5	6	3	4	7	6	3	8
Bay09	1	4	2	3	4	3	1	3	8	2
Bay10	4	8	1	4	7	2	3	4	4	6

Table 3: Bay-process assignments for 20-bay layouts

	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
Bay01	6	4	1	4	7	4	5	4	2	4
Bay02	1	1	1	8	1	4	3	2	3	1
Bay03	5	3	4	6	4	1	6	4	3	6
Bay04	4	7	6	4	1	1	1	4	6	1
Bay05	4	2	7	1	1	2	6	5	1	5
Bay06	2	3	8	1	3	3	8	7	7	1
Bay07	7	2	5	6	2	3	7	1	1	1
Bay08	3	4	6	4	6	7	6	8	7	6
Bay09	6	4	1	7	5	5	1	4	4	5
Bay10	7	1	1	7	6	1	3	3	4	3
Bay11	8	6	3	6	5	2	7	7	6	4
Bay12	1	6	4	4	6	6	4	1	5	8
Bay13	1	5	2	3	8	6	4	1	6	4
Bay14	5	8	5	5	4	7	4	6	1	2
Bay15	4	4	4	2	7	1	1	6	1	6
Bay16	3	5	3	3	4	6	5	1	4	4
Bay17	2	1	7	1	1	5	1	6	2	7
Bay18	4	7	6	2	3	4	2	2	5	3
Bay19	1	6	2	5	2	4	4	3	4	7
Bay20	6	1	4	1	4	8	2	5	8	2

between two adjacent bays to 1 because the simulation generator allows one shortcut. In practice, two or more shortcuts can exist between bays of large equipment.

We define problem instances by changing the value of η. We do not know how

large or small η should be, so we fix it to a certain value, define a single instance, and obtain a track design. We compare multiple designs from different η's and select the best track layout. In our study, $\eta \in \{0, 0.05, 0.1, 0.15, \ldots, 0.7\}$. The larger the value of η is, the more disjoint are the alternative paths. In some bay-process assignments, it happens that the optimization problem is infeasible. We set $\eta \leq 0.7$ in order to prevent this.

We report four performance metrics: number of completed transfer requests, time in system, delay ratio, and speed index.

Number of completed transfer requests We count completed transfers during 12,000 s after the warm-up period.

Time in system Time in system is the sum of waiting time until a vehicle is assigned, travel time for pickup, unloading time, travel time for delivery, and loading. Simply, it equals the delivery completion time minus the request generation time. We report the average over all transfer requests.

Delay ratio Delay ratio is the ratio of a transfer request's total travel time (pickup and delivery) to its uncongested travel time. We report the average over all transfer requests.

Speed index Speed index is a system-wise metric "at the moment". At every five seconds, we identify assigned vehicles that do not stop for unloading or loading (moving vehicles), their actual velocities, and their maximum available velocities on their current locations. For each moving vehicle, we calculate the ratio of its actual speed to its maximum available speed, and average the ratios over all moving vehicles. We report the average over 2,400 observations obtained during 12,000 s (12,000 s / 5 s = 2,400).

5.2 Base case results

In the base case results, we observe that the design from NDPA with a certain value of η outperforms the others and that η has significant relationship with routing performance.

We obtain the same track layout from several problem instances. They appear in neighbor values of η. Tables 4 and 5 present which layouts appear multiple times in 10-bay and 20-bay instances.

Table 4: Layouts appearing multiple times in 10-bay instances

η	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
0.00								a08		
0.05	a01				a05		a07	a08		
0.10	a01		a03		a05		a07			
0.15		a02	a03			a06	a07			
0.20	b01	a02		a04		a06			a09	
0.25	b01	a02		a04					a09	
0.30					b05	b06				a10
0.35					b05	b06		b08		a10
0.40			b03					b08		
0.45			b03		c05		b07			
0.50	c01				c05		b07		b09	
0.55	c01			b04		c06			b09	
0.60	c01			b04		c06				
0.65		b02	c03			c06		c08		
0.70		b02	c03					c08		

5.2.1 Overview

We present the computational results of three approaches proposed in Chapter 3:

- NDPA(η) is the track design obtained from NDPA with η, the uniqueness of alternative paths. NDPA(b) denotes the design where average time in system is minimized over all η's, so its η can vary based on bay-process assignments.
- ScenarioSUM relies on congestion scenarios. We designate the most congested track segment for each scenario and search for a design that is functional in

Table 5: Layouts appearing multiple times in 20-bay instances

η	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
0.00										
0.05		a02			a05				a09	
0.10	a01	a02			a05				a09	
0.15	a01		a03			a06			a09	
0.20			a03			a06		a08		
0.25				a04				a08		a10
0.30	b01			a04			a07			a10
0.35	b01		b03				a07			
0.40			b03					b08		
0.45		b02			b05			b08		
0.50		b02			b05			b08		
0.55		b02	c03			b06				b10
0.60			c03			b06			b09	b10
0.65	c01		c03	b04					b09	
0.70	c01			b04					b09	

every scenario. The objective is to minimize the sum of the weighted total travel time over all commodities.

- ScenarioMAX also depends on congestion scenarios. However, its objective function is to minimize the maximum of the weighted total travel time over all commodities.

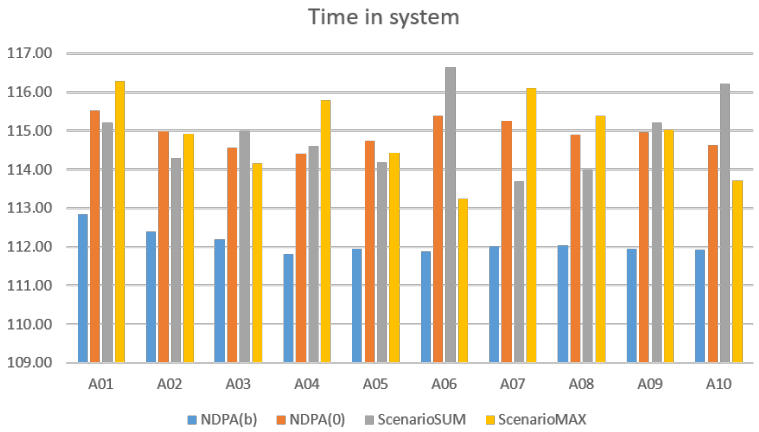
We plug three approaches into Algorithm 2 in Chapter 4. We assume four congestion scenarios for ScenarioSUM and ScenarioMAX, i.e., feasible designs are functional under four kinds of edge failures. To select the congested edge, we find a track segment with the largest delay ratio in the simulation, which is not a shortcut. If a selected track segment consists of multiple edges, then we select the first edge according to its direction.

Table 6 and Figures 17(a)-17(d) report the selected performance metrics. NDPA(b) is consistently better than other designs. On average, we observe 2.43%, 1.17%,

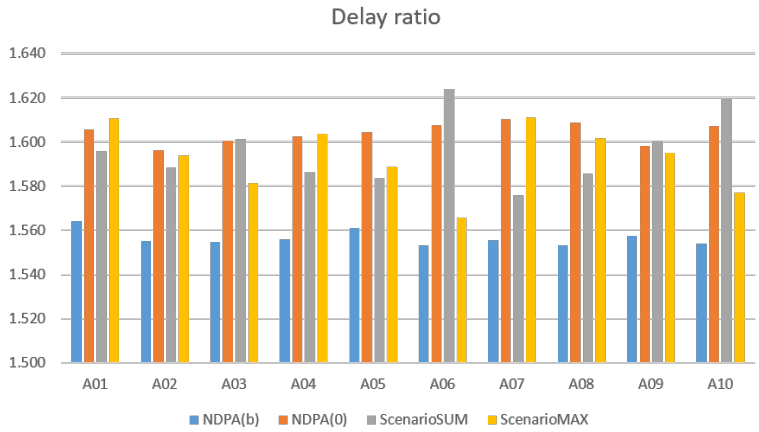
2.47%, and 3.29% improvements over NDPA(0) in time in system, number of completed requests, delay ratio, and speed index, respectively. We observe that NDPA(b) shows improvements over ScenarioSUM and ScenarioMAX but that ScenarioSUM and ScenarioMAX are not consistently superior to NDPA(0). In A04 and A09, their designs are worse than NDPA(0). We propose two reasons why the scenario-based approaches performed poorly. First, the number of scenarios and which edge to block in the optimization might not be adjusted carefully. Second, the constraints of ScenarioSUM or ScenarioMAX might be too conservative. Congestion delays vehicle movements, but it does not completely block traffic.

Table 6: Comparison with other approaches

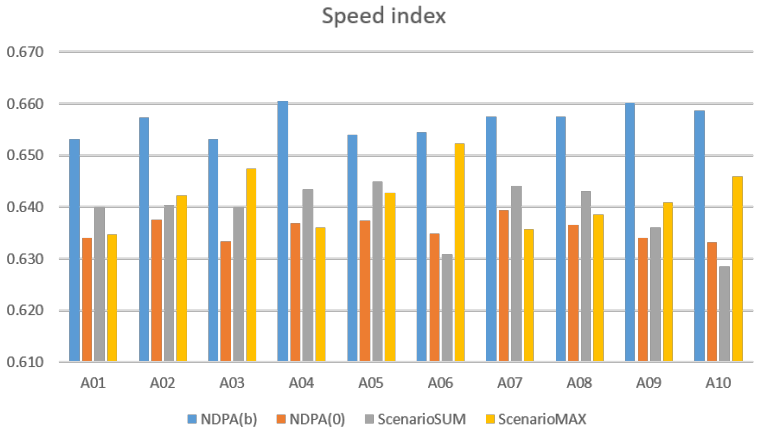
Time in system	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
NDPA(b)	112.85	112.40	112.20	111.80	111.95	111.87	112.01	112.04	111.95	111.93
NDPA(0)	115.52	114.99	114.57	114.41	114.74	115.39	115.25	114.89	114.96	114.62
ScenarioSUM	115.20	114.30	115.01	114.60	114.19	116.63	113.68	114.01	115.22	116.21
ScenarioMAX	116.29	114.92	114.17	115.80	114.43	113.24	116.10	115.39	115.02	113.72
Delay ratio	**A01**	**A02**	**A03**	**A04**	**A05**	**A06**	**A07**	**A08**	**A09**	**A10**
NDPA(b)	1.564	1.555	1.555	1.556	1.561	1.553	1.556	1.553	1.558	1.554
NDPA(0)	1.605	1.596	1.600	1.602	1.604	1.608	1.610	1.609	1.598	1.607
ScenarioSUM	1.596	1.589	1.601	1.586	1.583	1.624	1.576	1.585	1.600	1.619
ScenarioMAX	1.611	1.594	1.581	1.604	1.589	1.566	1.611	1.602	1.595	1.577
Speed index	**A01**	**A02**	**A03**	**A04**	**A05**	**A06**	**A07**	**A08**	**A09**	**A10**
NDPA(b)	0.653	0.657	0.653	0.661	0.654	0.655	0.657	0.657	0.660	0.659
NDPA(0)	0.634	0.638	0.633	0.637	0.637	0.635	0.639	0.637	0.634	0.633
ScenarioSUM	0.640	0.640	0.640	0.643	0.645	0.631	0.644	0.643	0.636	0.628
ScenarioMAX	0.635	0.642	0.647	0.636	0.643	0.652	0.636	0.638	0.641	0.646
Completed requests	**A01**	**A02**	**A03**	**A04**	**A05**	**A06**	**A07**	**A08**	**A09**	**A10**
NDPA(b)	14831.2	14831.7	14824.8	14872.8	14831.6	14836.5	14822.1	14859.6	14870.5	14851.1
NDPA(0)	14658.8	14662.7	14649.6	14664.8	14696.4	14654.5	14683.4	14693.3	14672.1	14636.0
ScenarioSUM	14661.9	14682.2	14599.6	14679.5	14725.6	14637.1	14782.6	14711.0	14657.6	14584.3
ScenarioMAX	14627.0	14670.4	14770.0	14651.1	14700.1	14791.0	14657.1	14621.5	14655.7	14752.4



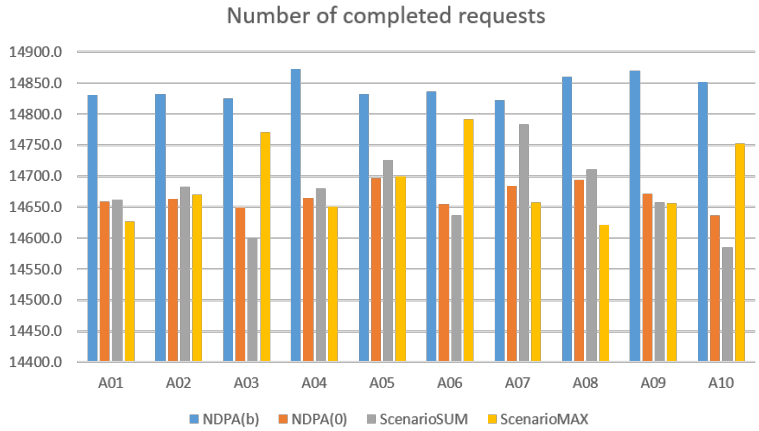
(a) Time in system



(b) Delay ratio



(c) Speed index



(d) Number of completed requests

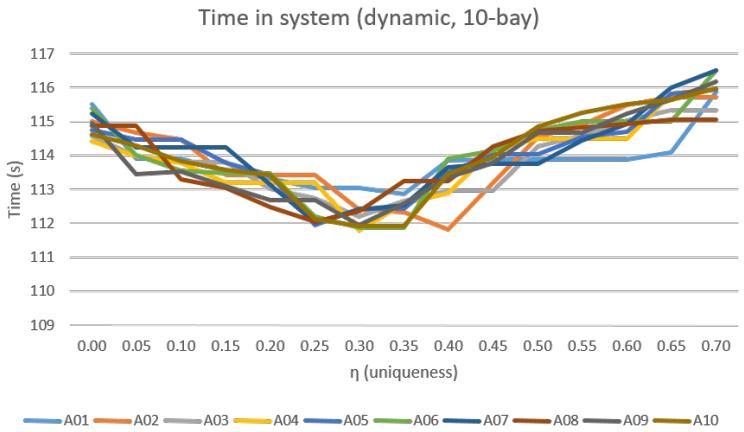
Figure 17: Comparison with other routing approaches

5.2.2 Relationship between η and performance metrics

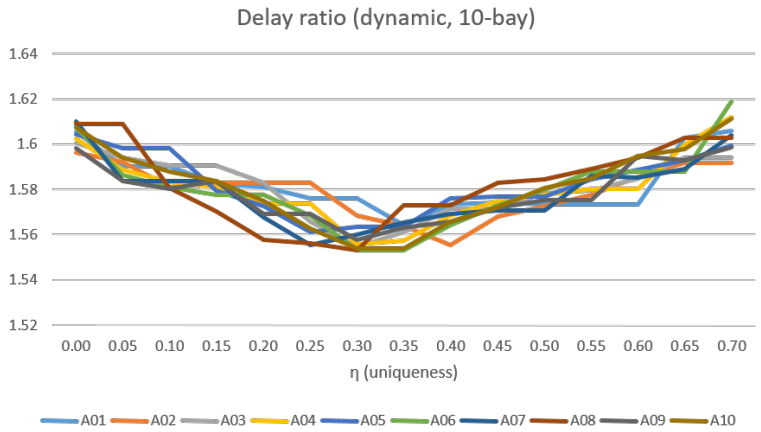
Focusing on NDPA(η), we investigate the relationship between η and performance metrics. Figures 18(a)-18(d) report four performance metrics of the designs from $\eta = 0, 0.05, \ldots, 0.7$. We observe the relationship between metrics and η. Specifically, time in system, delay ratio, and speed index form V-shaped curves as η increases. They achieve their minima when η is close to 0.3 though specific values of η vary based on bay-process assignments. The relationship between the number of completed requests and η turns over. We observe similar patterns in 20-bay instances in Figures 19(a)-19(d).

Table 7 shows the averages and 95% confidence intervals of selected performance metrics from one bay-process assignment. Assignment A01 attains the best layout when $\eta = 0.35$. The superscripts of some η's mean that have the same design. The solution designs remain the same when η equals 0.05 and 0.1, 0.2 and 0.25, or 0.5, 0.55, and 0.6.

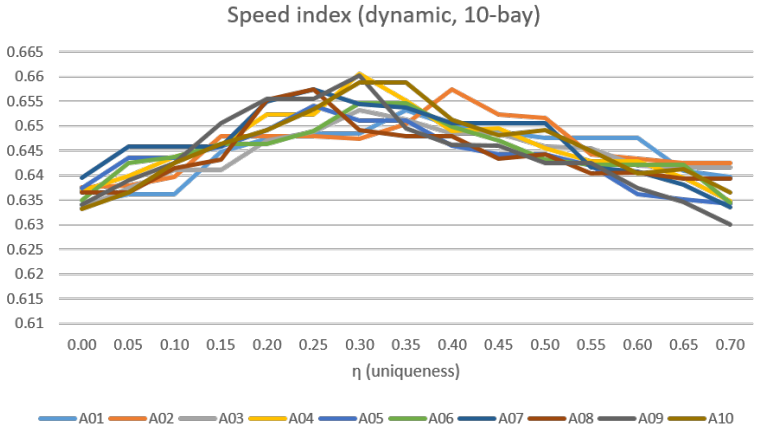
Tables 8 and 9 list the results of paired t-tests for time in system and number of completed requests. Paired t-test statistics have 19 degrees of freedom from 20 replications and critical values of 2.09 for 95% two-sided tests. NDPA(b), which comes from $\eta = 0.35$ for assignment A01, provides a significant decrease of time in system, and a significant increase of number of completed requests compared to most of the other designs.



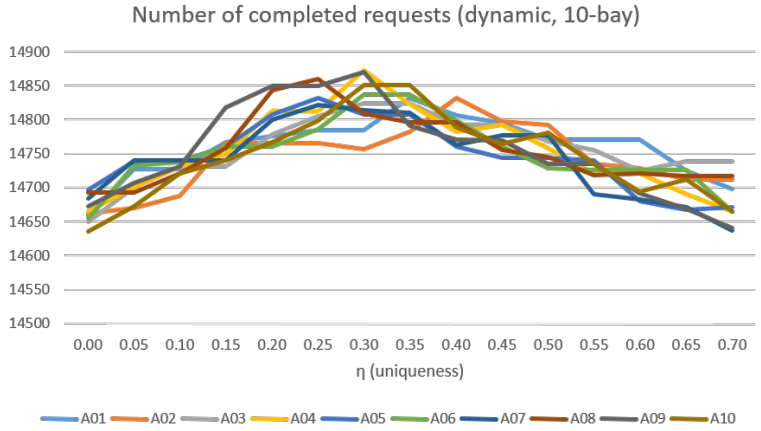
(a) Time in system



(b) Delay ratio

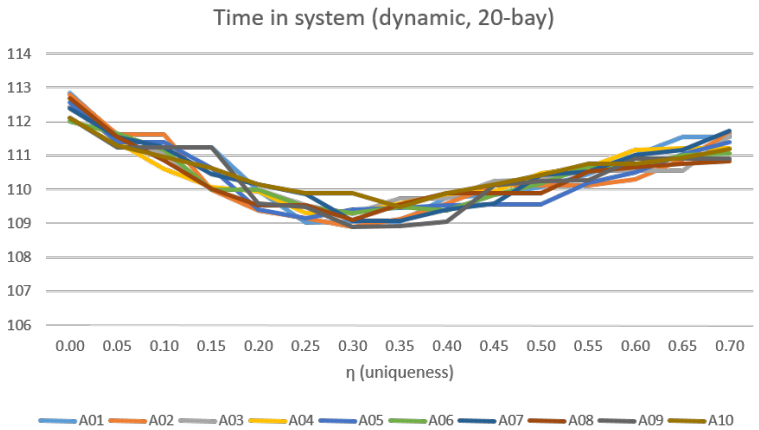


(c) Speed index

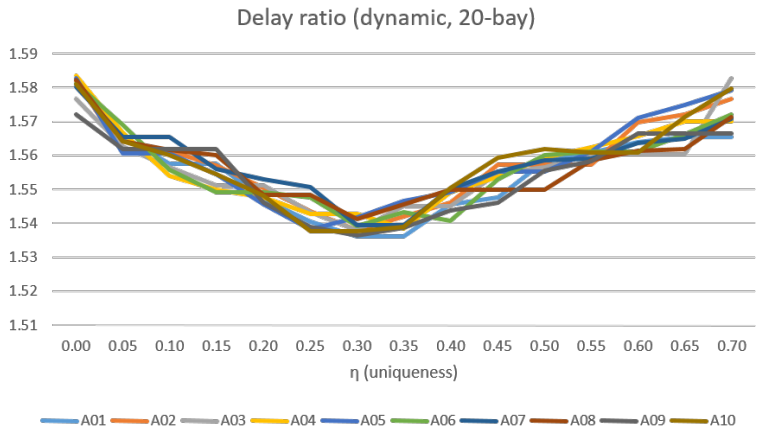


(d) Number of completed requests

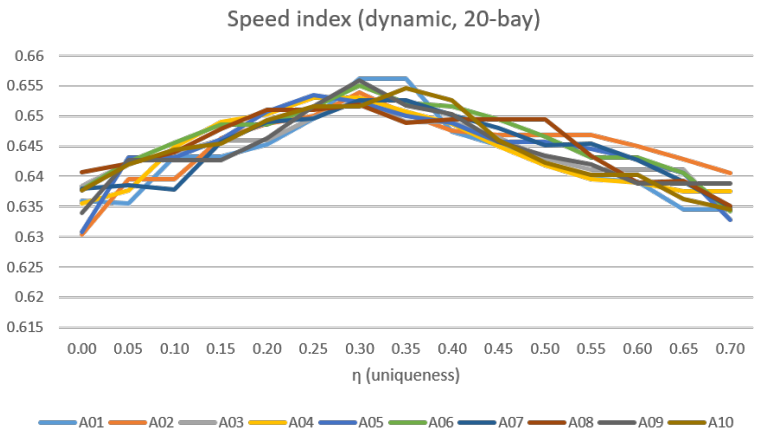
Figure 18: Base case results (10-bay, dynamic)



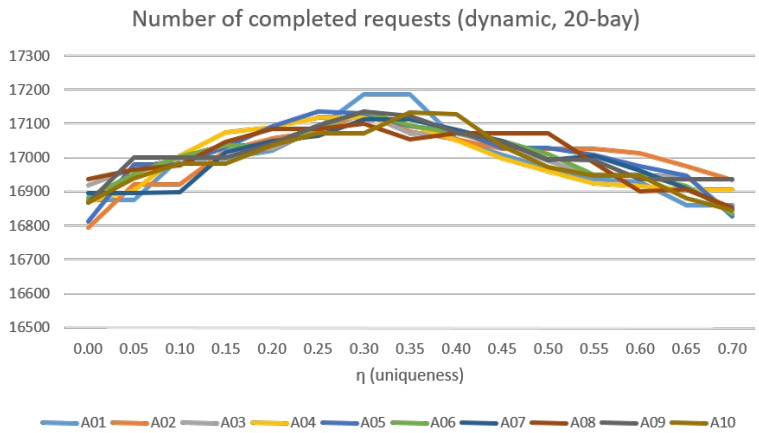
(a) Time in system



(b) Delay ratio



(c) Speed index



(d) Number of completed requests

Figure 19: Base case results (20-bay, dynamic)

Table 7: Performance metrics of a 10-bay instance (A01)

η	Number of completed requests	Time in system		Delay ratio		Speed index	
		Average	C.I.	Average	C.I.	Average	C.I.
0.00	14658.8	115.52	(114.61, 116.44)	1.605	(1.595, 1.616)	0.634	(0.630, 0.639)
0.05[a]	14727.7	113.92	(112.97, 114.86)	1.590	(1.582, 1.598)	0.636	(0.631, 0.641)
0.10[a]	14727.7	113.92	(112.97, 114.86)	1.590	(1.582, 1.598)	0.636	(0.631, 0.641)
0.15	14767.3	113.52	(112.51, 114.53)	1.583	(1.574, 1.591)	0.645	(0.640, 0.649)
0.20[b]	14775.4	113.32	(112.29, 114.35)	1.581	(1.569, 1.593)	0.647	(0.644, 0.651)
0.25[b]	14784.1	113.04	(112.11, 113.96)	1.576	(1.565, 1.588)	0.648	(0.643, 0.654)
0.30	14784.1	113.04	(112.11, 113.96)	1.576	(1.565, 1.588)	0.648	(0.643, 0.654)
0.35	14831.2	112.85	(111.84, 113.87)	1.564	(1.553, 1.575)	0.653	(0.648, 0.658)
0.40	14806.0	113.86	(112.94, 114.78)	1.573	(1.566, 1.581)	0.650	(0.644, 0.656)
0.45	14795.4	113.88	(112.97, 114.78)	1.574	(1.566, 1.582)	0.649	(0.643, 0.655)
0.50[c]	14770.7	113.89	(112.84, 114.94)	1.574	(1.565, 1.582)	0.648	(0.643, 0.652)
0.55[c]	14770.7	113.89	(112.84, 114.94)	1.574	(1.565, 1.582)	0.648	(0.643, 0.652)
0.60[c]	14770.7	113.89	(112.84, 114.94)	1.574	(1.565, 1.582)	0.648	(0.643, 0.652)
0.65	14724.5	114.08	(113.05, 115.10)	1.603	(1.592, 1.613)	0.641	(0.635, 0.647)
0.70	14698.1	115.86	(114.81, 116.91)	1.606	(1.594, 1.618)	0.640	(0.635, 0.645)

Table 8: Paired t-test statistics of 10-bay instances (A01, time in system)

η	0.00	0.05[a]	0.10[a]	0.15	0.20	0.25[b]	0.30[b]	0.35	0.40	0.45	0.50[c]	0.55[c]	0.60[c]	0.65	0.70
0.00		**2.90**	**2.90**	**3.92**	**4.44**	**5.01**	**5.01**	**5.63**	**3.33**	**3.20**	**2.90**	**2.90**	**2.90**	-0.09	-0.57
0.05[a]			0.00	0.81	1.26	1.86	1.86	**2.36**	0.12	0.08	0.05	0.05	0.05	**-3.01**	**-3.41**
0.10[a]				0.81	1.26	1.86	1.86	**2.36**	0.12	0.08	0.05	0.05	0.05	**-3.01**	**-3.41**
0.15					0.47	1.14	1.14	1.68	-0.80	-0.81	-0.74	-0.74	-0.74	**-4.05**	**-4.44**
0.20						0.69	0.69	1.24	-1.33	-1.31	-1.18	-1.18	-1.18	**-4.58**	**-4.95**
0.25[b]							0.00	0.50	-2.02	-1.97	-1.76	-1.76	-1.76	**-5.15**	**-5.50**
0.30[b]								0.50	-2.02	-1.97	-1.76	-1.76	-1.76	**-5.15**	**-5.50**
0.35									**-2.65**	**-2.56**	**-2.24**	**-2.24**	**-2.24**	**-5.78**	**-6.12**
0.40										-0.04	-0.06	-0.06	-0.06	**-3.46**	**-3.88**
0.45											-0.02	-0.02	-0.02	**-3.32**	**-3.74**
0.50[c]												0.00	0.00	**-3.01**	**-3.41**
0.55[c]													0.00	**-3.01**	**-3.41**
0.60[c]														**-3.01**	**-3.41**
0.65															-0.49
0.70															

Table 9: Paired t-test statistics of 10-bay instances (A01, number of completed requests)

η	0.00	0.05[a]	0.10[a]	0.15	0.20	0.25[b]	0.30[b]	0.35	0.40	0.45	0.50[c]	0.55[c]	0.60[c]	0.65	0.70
0.00		-2.03	-2.03	**-3.25**	**-3.50**	**-3.73**	**-3.73**	**-5.19**	**-4.47**	**-4.08**	**-3.35**	**-3.35**	**-3.35**	-1.91	-1.14
0.05[a]			0.00	-1.20	-1.45	-1.70	-1.70	**-3.15**	**-2.41**	-2.05	-1.30	-1.30	-1.30	0.09	0.87
0.10[a]				-1.20	-1.45	-1.70	-1.70	**-3.15**	**-2.41**	-2.05	-1.30	-1.30	-1.30	0.09	0.87
0.15					-0.25	-0.52	-0.52	-1.98	-1.21	-0.87	-0.11	-0.11	-0.11	1.28	2.07
0.20						-0.27	-0.27	-1.74	-0.96	-0.62	0.14	0.14	0.14	1.52	**2.32**
0.25[b]							0.00	-1.45	-0.68	-0.34	0.41	0.41	0.41	1.77	**2.55**
0.30[b]								-1.45	-0.68	-0.34	0.41	0.41	0.41	1.77	**2.55**
0.35									0.79	1.11	1.87	1.87	1.87	**3.20**	**4.00**
0.40										0.33	1.10	1.10	1.10	**2.47**	**3.27**
0.45											0.76	0.76	0.76	**2.11**	**2.90**
0.50[c]												0.00	0.00	1.38	**2.17**
0.55[c]													0.00	1.38	**2.17**
0.60[c]														1.38	**2.17**
0.65															0.77
0.70															

Congestion causes more frequent rerouting movements, but vehicles stay on the shortest paths rather than take alternative paths unless they are expected to be faster. We expect alternative paths to be disjoint with the shortest path while they are not excessively longer than the shortest path. A higher η controls the uniqueness of alternative paths but lengthens them. The relationships between rerouting frequencies and traffic conditions are not straightforward. For example, good alternative paths encourage rerouting, so they are essential to improve traffic conditions, but rerouting frequencies decrease when traffic conditions are fine.

Figures 20 and 21 illustrate the weighted averages of rerouting frequencies over all commodities in the 10-bay and 20-bay instances. Note that the weight of each commodity is its demand. If we ignore $\eta = 0$, then curves are M-shaped in most bay-process assignments. Either small or large value of η generates ill-conditioned alternative paths. When η equals 0.25 or 0.3, where routing performance is improved, rerouting frequencies become smallest.

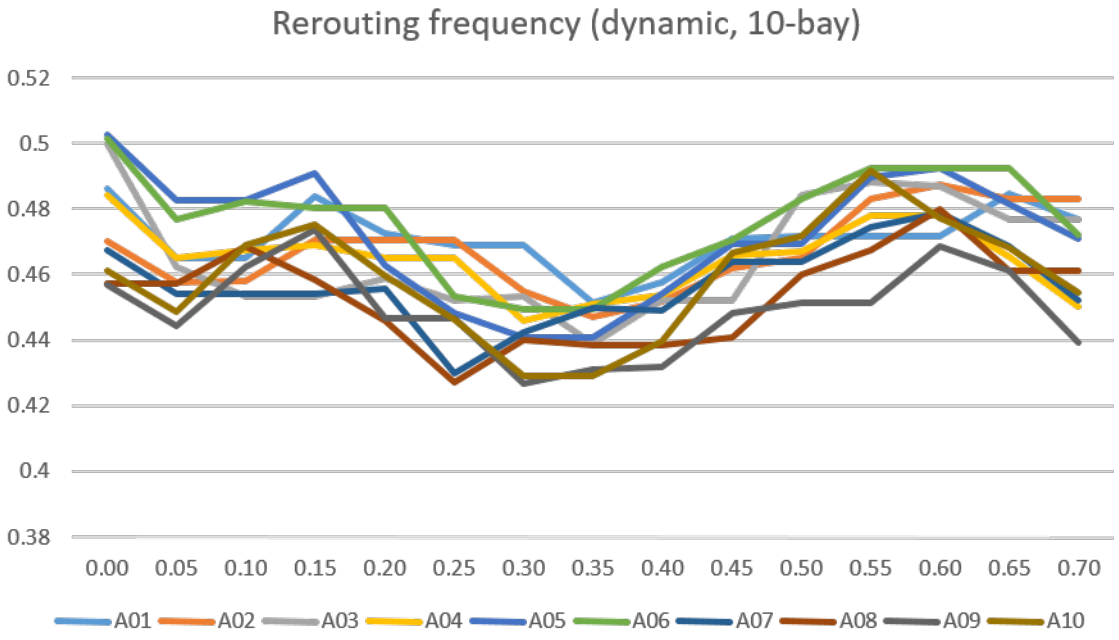


Figure 20: Rerouting frequency (base, 10-bay)

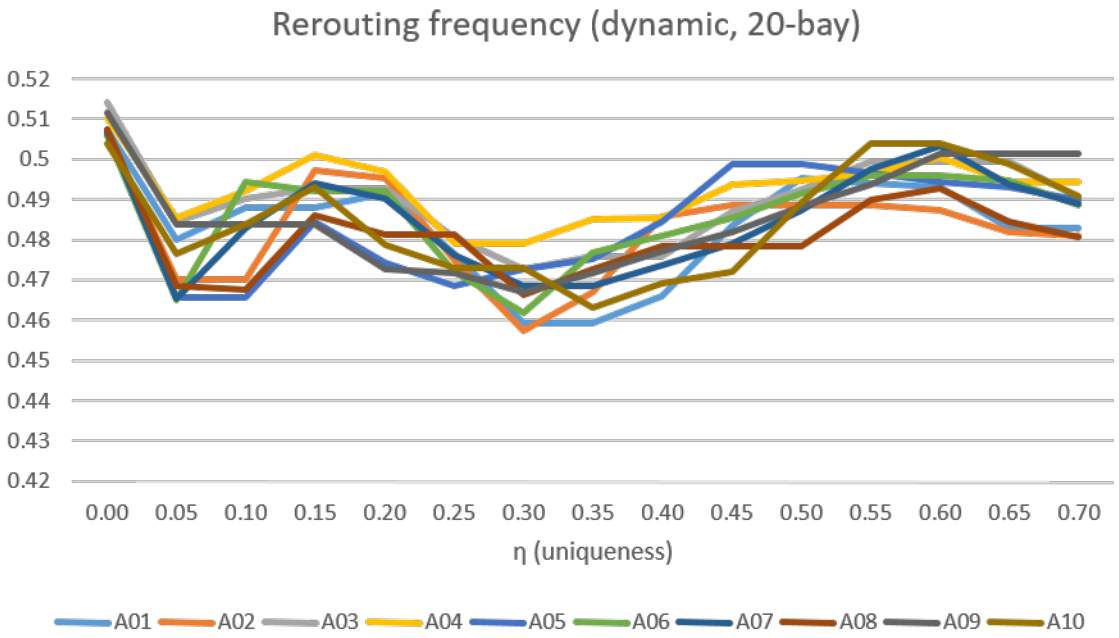


Figure 21: Rerouting frequency (base, 20-bay)

5.2.3 Characteristics of shortcuts in NDPA(b)

We observe that the direction of a shortcut between two bays depends on the demand difference of the bays. Let a shortcut be located on the track segment, say r_e, connecting the exit of bay A and the entrance of bay B. We calculate the demand sum of commodities that have origins at bay A, say d_A. Similarly, let d_B be the demand sum of commodities that have destinations at bay B. If $d_A > d_B$, then the shortcut has its tail on r_e. That is, the shortcut direction is set to provide a detour to the center or outer loop for commodities from bay A as early as possible. Otherwise, the shortcut has its head on r_e so that commodities can enter bay B via an alternative path through the shortcut.

Figure 22 shows the shortcut placements of NDPA(b) for bay-process assignment A01. In this layout, all the shortcuts between bays satisfy the relationship. For example, the demand sum of commodities from bay 1 is 0.158 while the demand sum of commodities to bay 2 is 0.013. This determines the direction of the red-circled

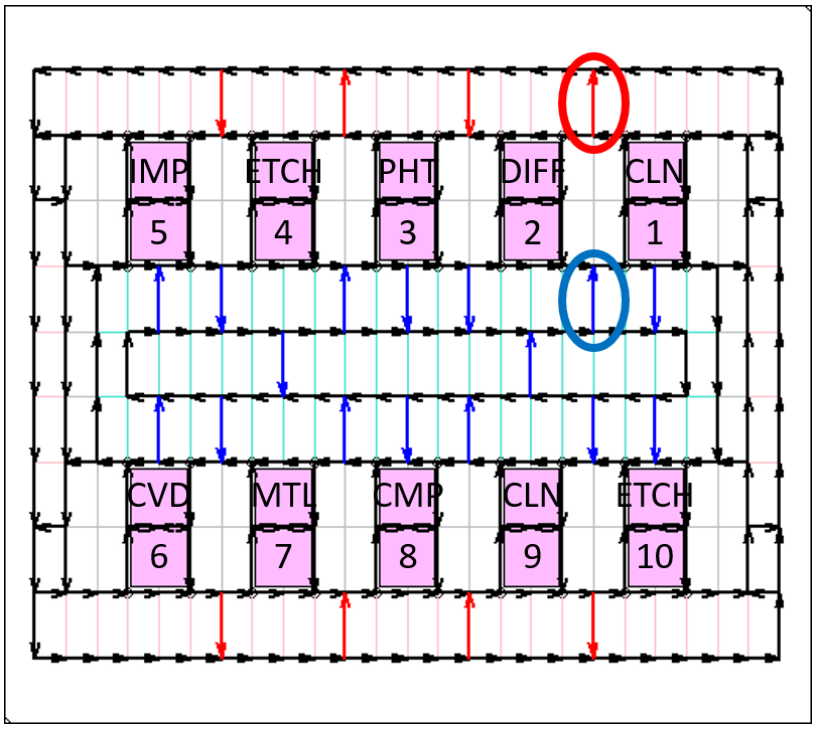


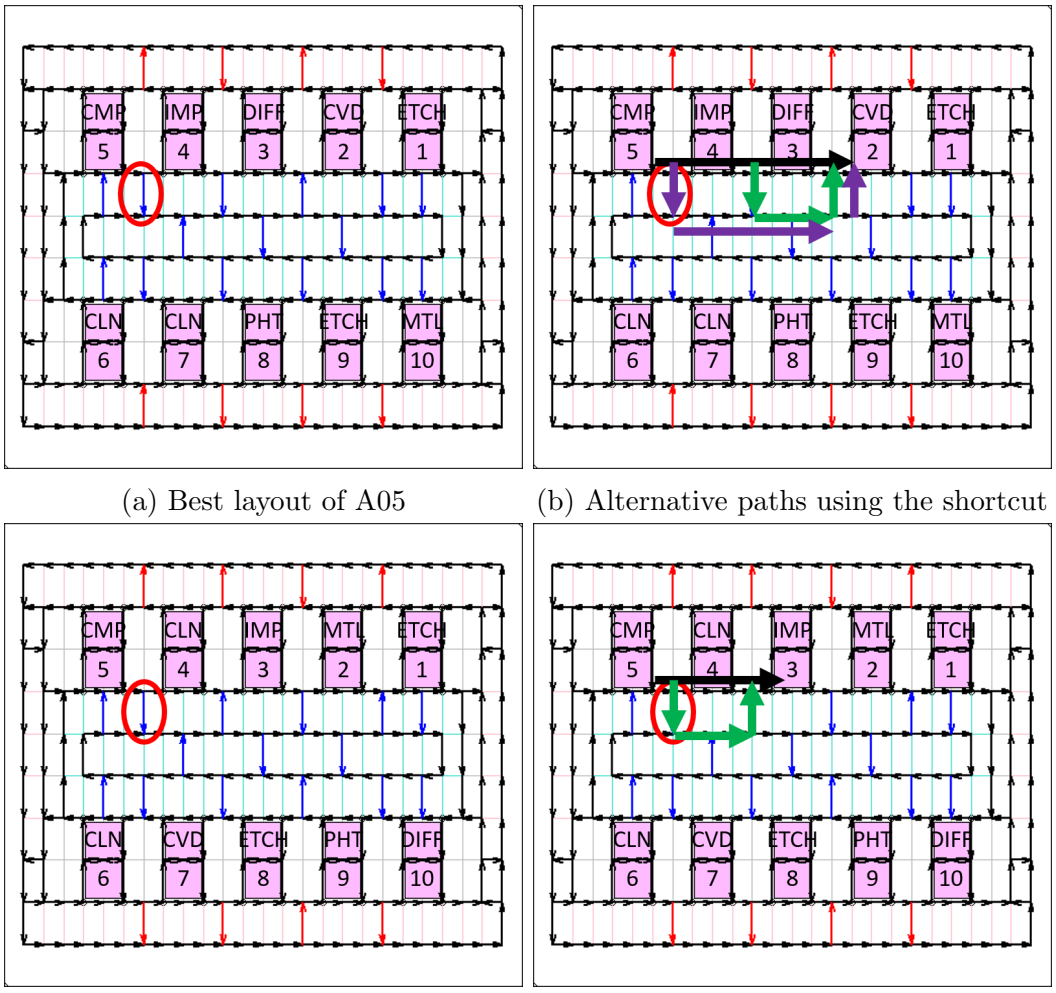
Figure 22: Best layout of A01

shortcut. On the other hand, the demand sum of commodities from bay 2 is 0.013, and the demand sum of commodities to bay 1 is 0.158. Then, this determines the direction of the blue-circled shortcut. Table 10 describes these relationships. Column 1 indicates the loop to which each track segment between bays are connected, by the shortcut on it. Column 6 identifies whether the node on the track segment between two bays is the tail or head of a shortcut, if the shortcut is placed.

Table 10: Predicted and actual shortcut directions

To Which Loop	Origin Bay (A)	Destination Bay (B)	$\sum_{k \in K: o_k = A} w_k$	$\sum_{k \in K: d_k = B} w_k$	Shortcut Node	Predicted Direction	Selected Direction
Outer Loop	1	2	0.158	0.013	Tail	↑	↑
	2	3	0.013	0.107	Head	↓	↓
	3	4	0.107	0.052	Tail	↑	↑
	4	5	0.052	0.243	Head	↓	↓
Outer Loop	6	7	0.075	0.017	Tail	↓	↓
	7	8	0.017	0.124	Head	↑	↑
	8	9	0.124	0.158	Head	↑	↑
	9	10	0.158	0.052	Tail	↓	↓
Center Loop	5	4	0.243	0.052	Tail	↓	↓
	4	3	0.052	0.107	Head	↑	↑
	3	2	0.107	0.013	Tail	↓	↓
	2	1	0.013	0.158	Head	↑	↑
Center Loop	10	9	0.052	0.158	Head	↓	↓
	9	8	0.158	0.124	Tail	↑	↑
	8	7	0.124	0.017	Tail	↑	↑
	7	6	0.017	0.075	Head	↓	↓

We also observe exceptions, which provide better alternative paths to commodities with high demand. In Figure 23, shortcuts violating the relation are circled in red. The shortcuts close to bay 5 in Figures 23(a) and 23(c) do not satisfy the relationship. Figure 23(b) shows that the red-circled shortcut assures the purple-colored alternative path to the commodity from CMP (bay 5) to CVD (bay 2), which has the third-highest demand among all commodities. Figure 23(d) shows that the shortcut provides the green-colored alternative path to the commodity from bay 5 (CMP) to bay 3 (IMP), which has the highest demand.



(a) Best layout of A05 (b) Alternative paths using the shortcut

(c) Best layout of A06 (d) Alternative paths using the shortcut

Figure 23: Impacts of shortcuts on alternative paths

These observations cannot explain all of the shortcut placements. For example,

the shortcuts between the entrance and exit of bays are not always placed. The best track layout needs to consider all transfer requests, which eventually motivates a global approach similar to our proposed optimization problem.

We report observations regarding the alternative paths in the track layouts obtained from NDPA. When the AMHS dispatches a vehicle to a path that is not the uncongested shortest path, we calculate the ratio of its uncongested travel time to the uncongested travel time of the shortest path. We report AP/SP, the average of the ratios. Note that the selected alternative paths can be longer than the second-shortest path. Figure 24 shows that AP/SP tends to increase when η increases but that the track layouts from the same η have different values of AP/SP based on bay-process assignments.

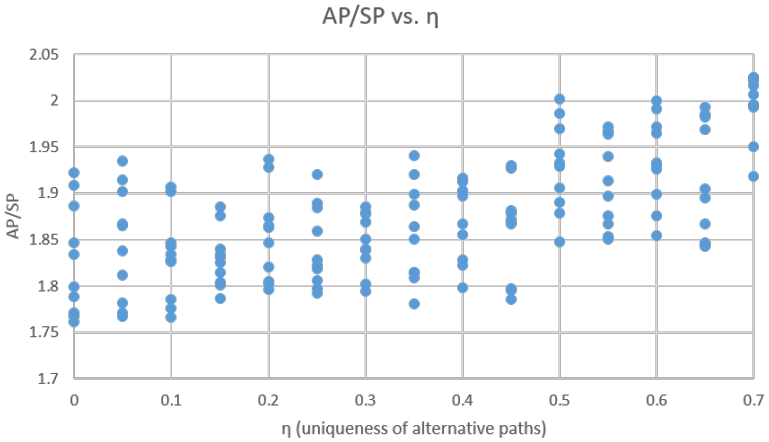


Figure 24: AP/SP vs. η

Figure 25 shows the relationship between AP/SP and the average delay ratio. The red circles indicate NDPA(b), the best layout for each bay-process assignment. We observe that the delay ratio increases as AP/SP increases or decreases becomes larger or smaller than, in our results, 1.85. As mentioned in Chapter 1, bad alternative paths are much longer than the uncongested shortest path (high AP/SP). Alternatively, they are not too long but do share many track segments with the uncongested shortest path (low AP/SP).

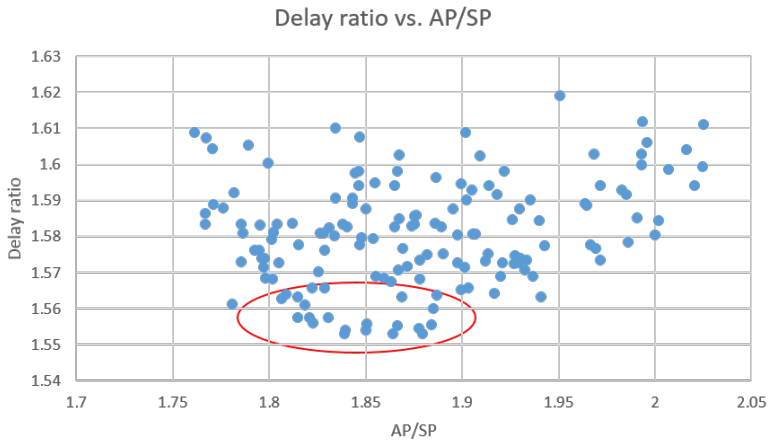


Figure 25: Delay ratio vs. AP/SP

To see if this relationship holds in general, we add the results from 100 random designs. In Figure 26, the observations define a region the boundary of which is a red-dashed line. Note that most of the newly added results are located above than the results of NDPA. Figures 27(a)- 27(d) report two other metrics, time in system and speed index, for the track layouts obtained from NDPA and 100 random designs.

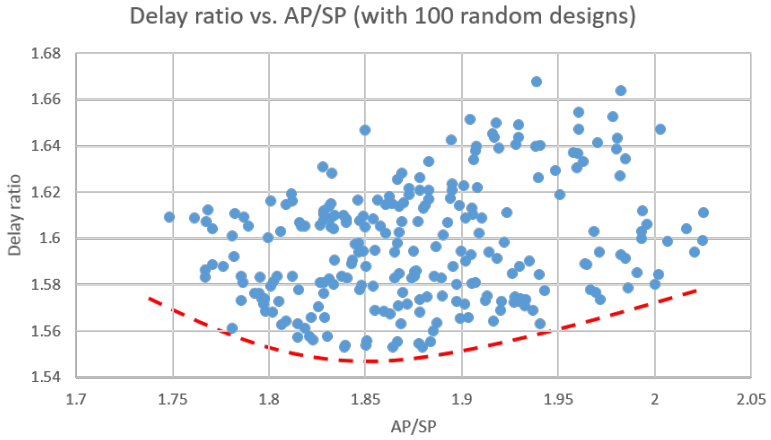
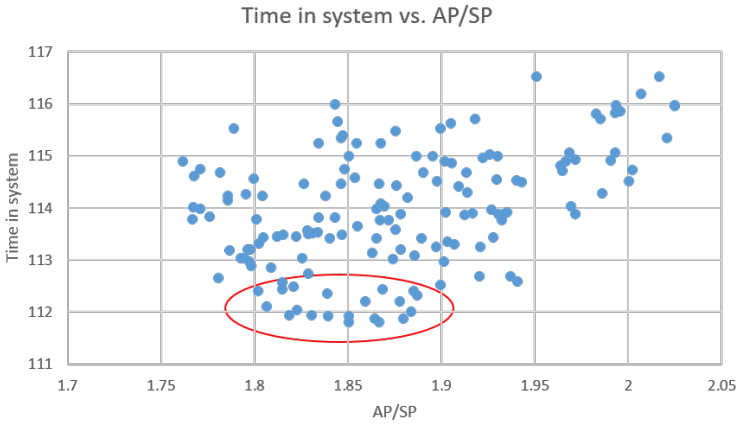
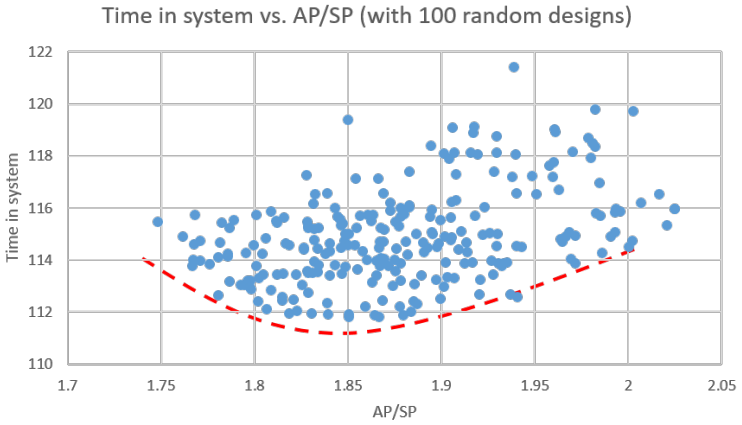


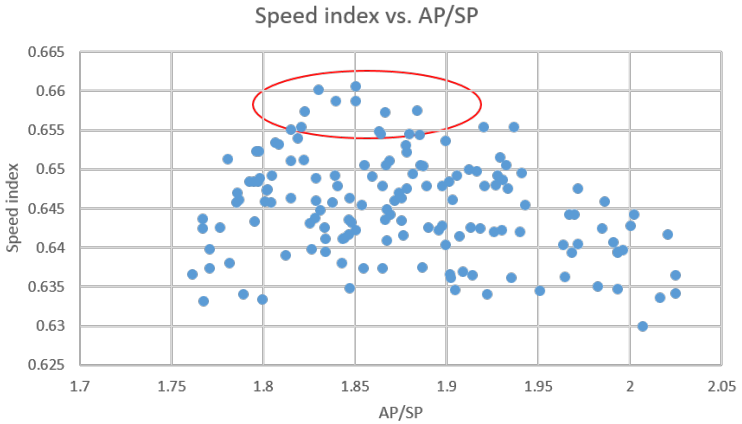
Figure 26: Delay ratio vs. AP/SP (with 100 random designs)



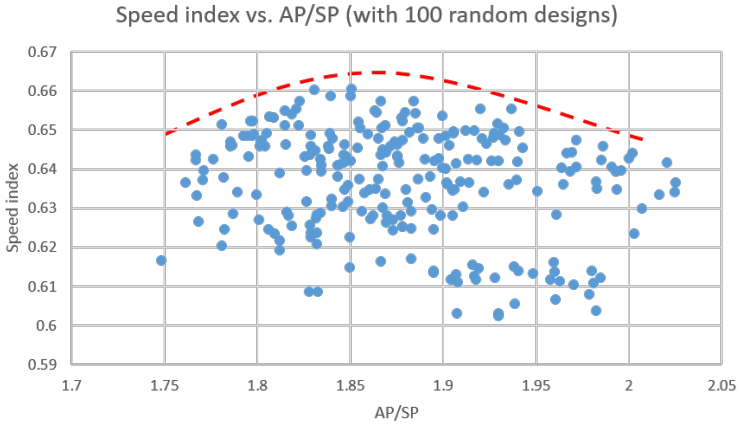
(a) Time in system vs. AP/SP



(b) Time in system vs. AP/SP (with 100 random designs)



(c) Speed index vs. AP/SP



(d) Speed index vs. AP/SP (with 100 random designs)

Figure 27: AP/SP and performance metrics

5.3 Sensitivity analysis

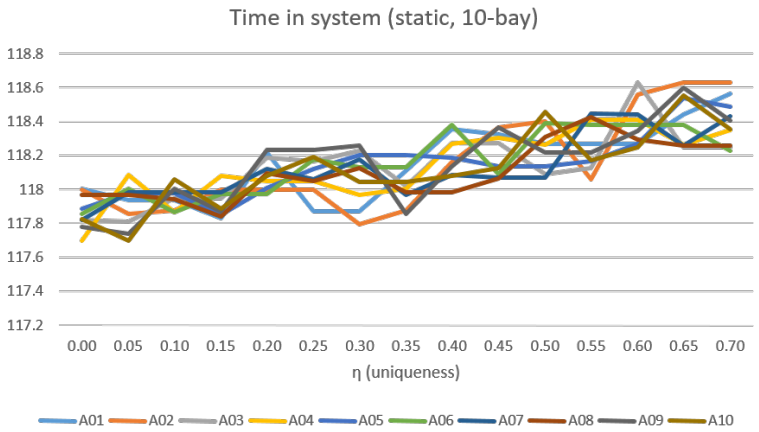
5.3.1 Sensitivity to routing schemes

As mentioned in Chapter 1, most fab AMHSs do not use dynamic routing. Therefore, we examine the performance of two other routing approaches in the track layouts obtained from NDPA:

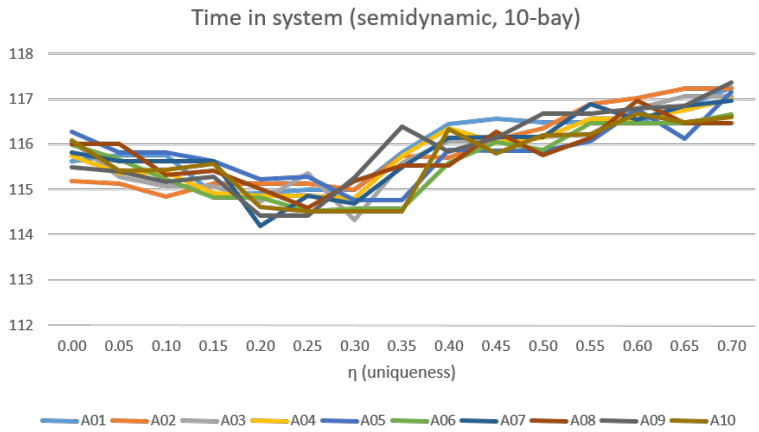
Static routing Travel time of a track segment is estimated a priori and does not change while in operation or simulation. Vehicles do not change their paths.

Semidynamic routing The AMHS updates travel time data periodically and searches for paths of transfer requests. Vehicles do not change their paths while they are being assigned. Although it does not reroute vehicles, semidynamic routing is more responsive to congestion than static routing.

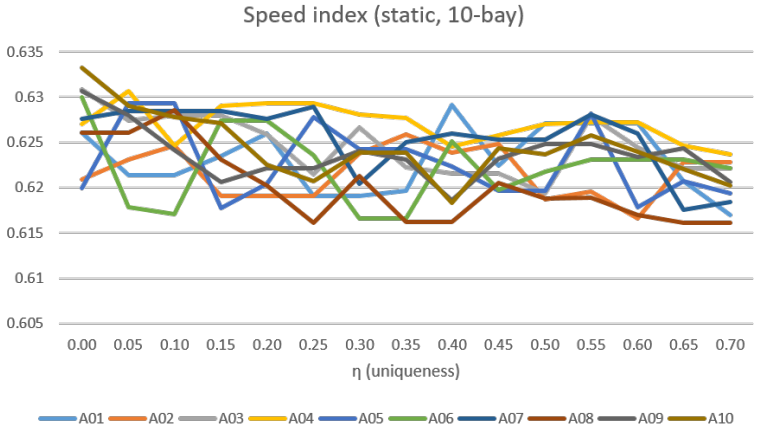
Figures 28(a)-28(d) report the performance metrics of different routing schemes in the 10-bay layouts we obtain in Section 5.2. Figures 29(a)-29(d) report the 20-bay results. In static routing, both time in system and speed index tend to worsen as η increases, but the difference of metrics between two layouts from $\eta = 0$ and $\eta = 0.7$ is not statistically significant. On the other hand, the results of semidynamic routing form V-shaped curves between time in system and η, but we do not observe significant improvements.



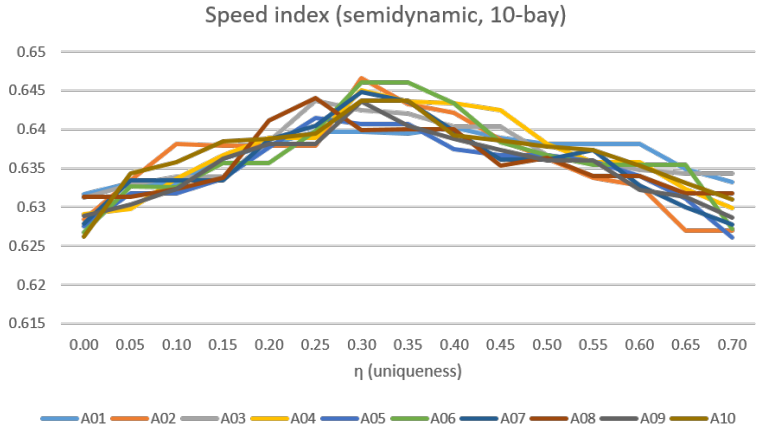
(a) Time in system (base, 10-bay, static)



(b) Time in system (base, 10-bay, semidynamic)

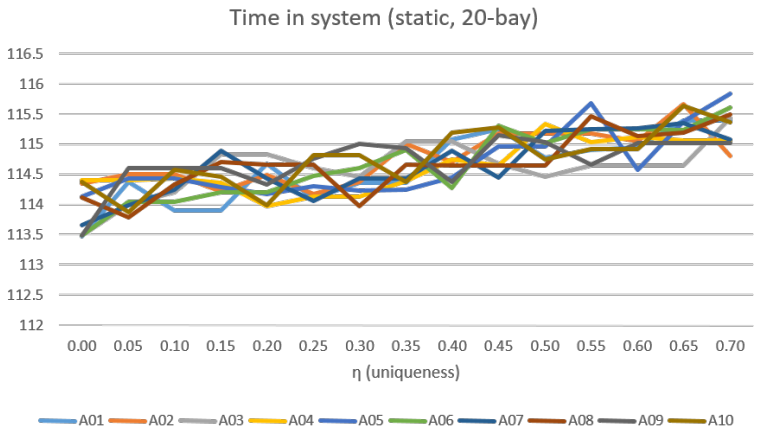


(c) Speed index (base, 10-bay, static)

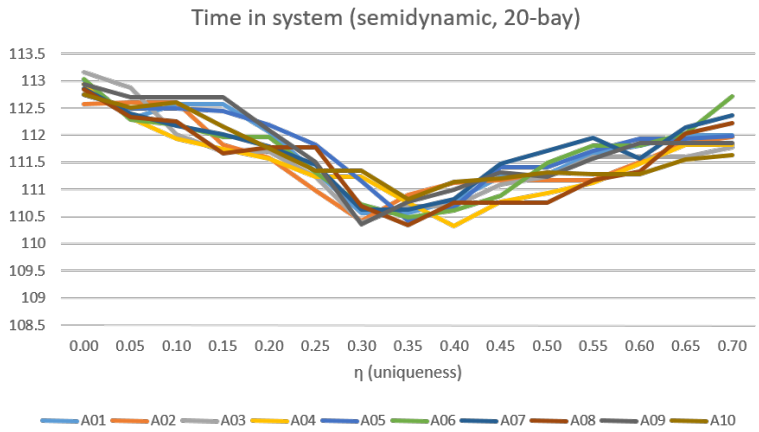


(d) Speed index (base, 10-bay, semidynamic)

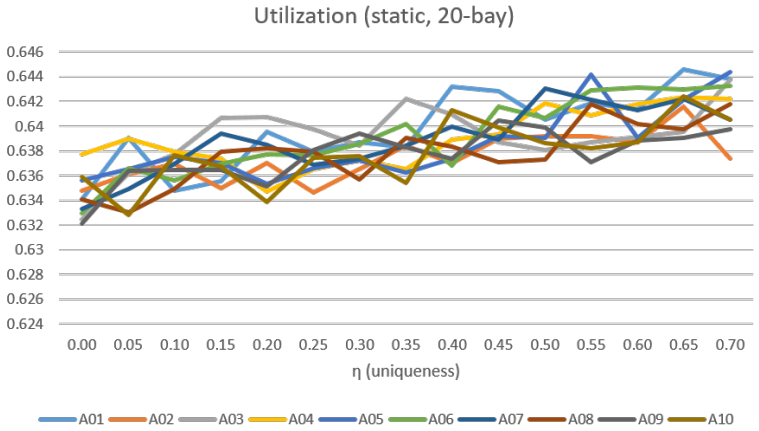
Figure 28: Routing performance of static and semidynamic routing (10-bay)



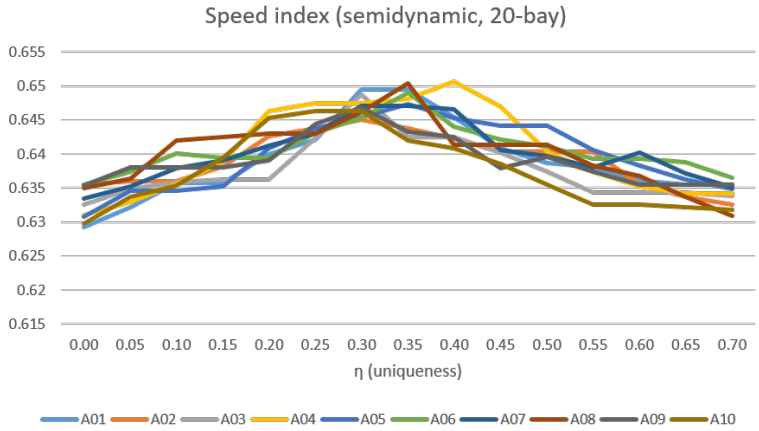
(a) Time in system (base, 20-bay, static)



(b) Time in system (base, 20-bay, semidynamic)



(c) Speed index (base, 20-bay, static)



(d) Speed index (base, 20-bay, semidynamic)

Figure 29: Routing performance of static and semidynamic routing (20-bay)

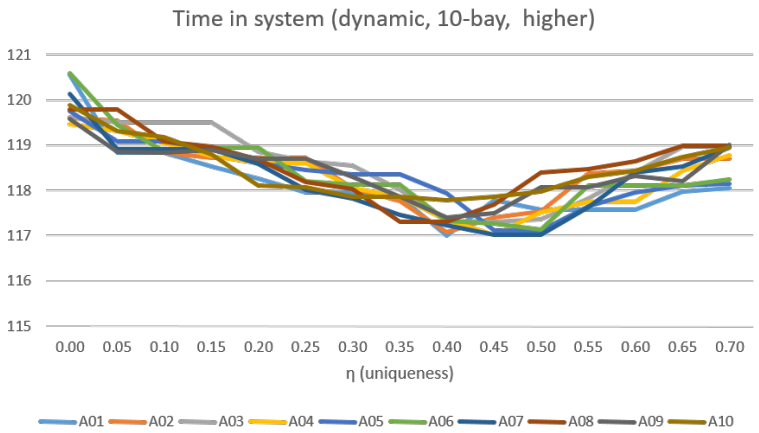
Table 11: Frequency of deadlock

	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
10-bay	0%	15%	10%	5%	0%	10%	5%	10%	10%	5%
20-bay	5%	5%	0%	0%	5%	0%	10%	0%	0%	5%

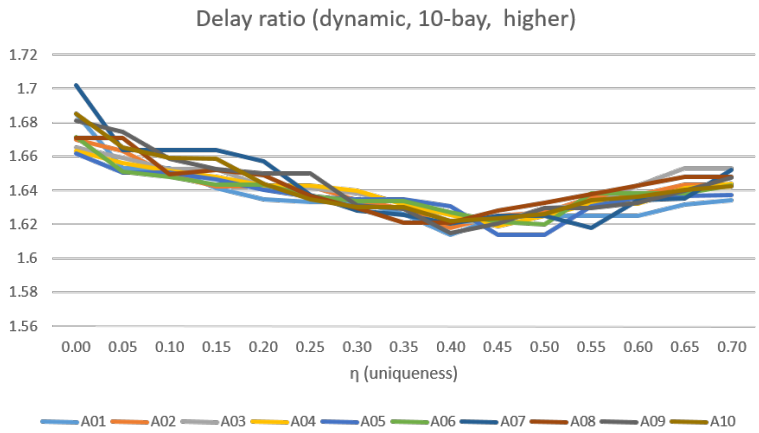
5.3.2 Sensitivity to higher workload

We also examine the robustness of the track layouts, NDPA(η), $\eta = 0, 0.05, \ldots, 0.7$ under higher workload. We know that the number of assigned vehicles, which have to stop for loading or unloading FOUP cartridges, increases when the average time between the transfer requests decreases. We also know that more stopping vehicles will aggravate traffic conditions. Although dynamic routing allows vehicles to evade congested locations, routing performance may worsen because of the stopping vehicles. In addition, serious deadlock may occur, which cannot be resolved by dynamic routing. Table 11 lists the frequency of deadlock under higher workload in 20 replications. 10-bay instances encounter deadlock more often than 20-bay instances. We suspect that the number of vehicles compared to the fab size is relatively large.

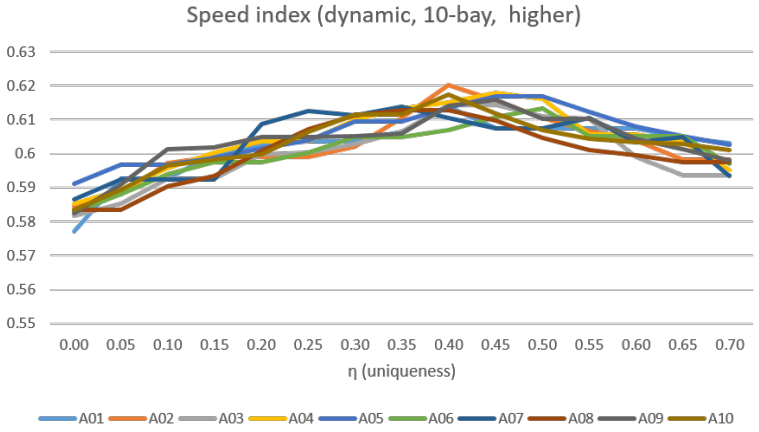
Figures 30(a)-30(d) and 31(a)-31(d) present performance metrics under higher workload. For each bay-process assignment, we collect the results from all replications without deadlock because the number of random seeds without deadlock in our bay-process assignments is small. The relationship between η and metrics is similar to the base case results, but each metric attains its best value at larger η under higher workload. We obtain the best layout when $\eta \approx 0.3$. Under higher workload, the layout when $\eta \approx 0.4$ shows better performance than NDPA(b). Based on this observation, we recommend selecting the layout from the largest η if several designs show similar results.



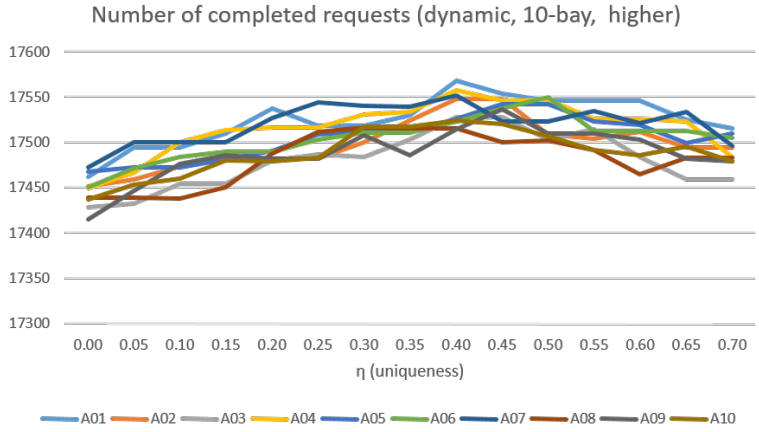
(a) Time in system



(b) Delay ratio

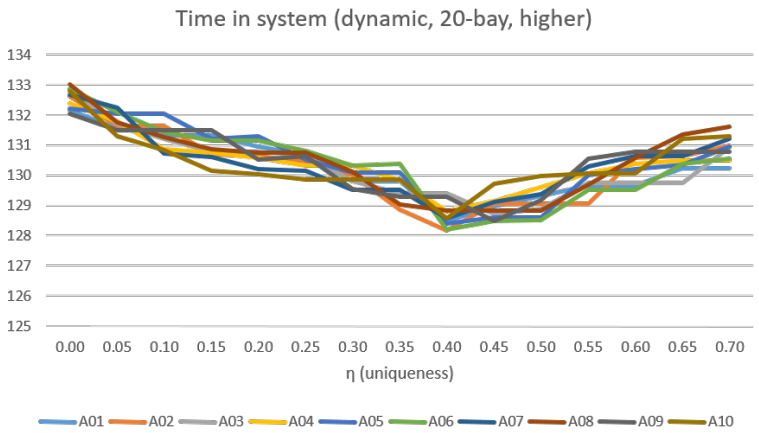


(c) Speed index

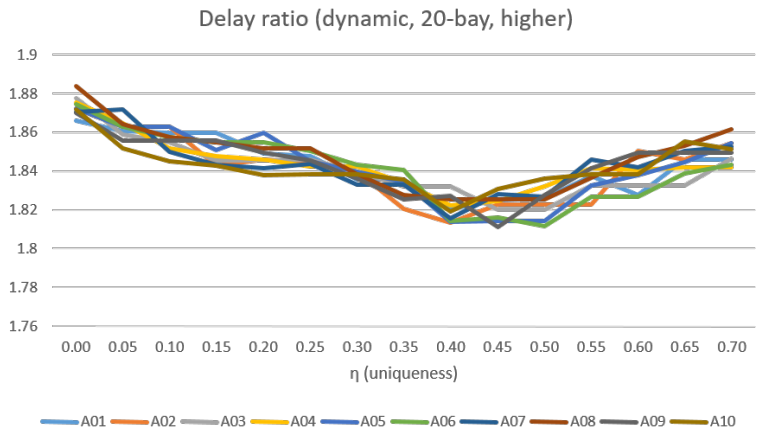


(d) Number of completed requests

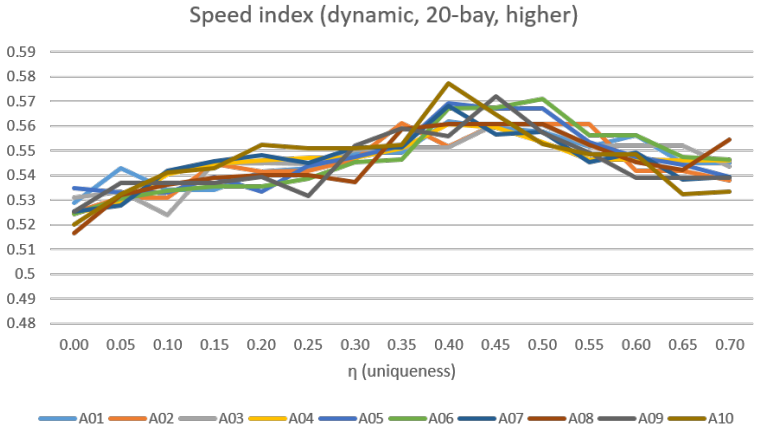
Figure 30: Higher workload results (10-bay)



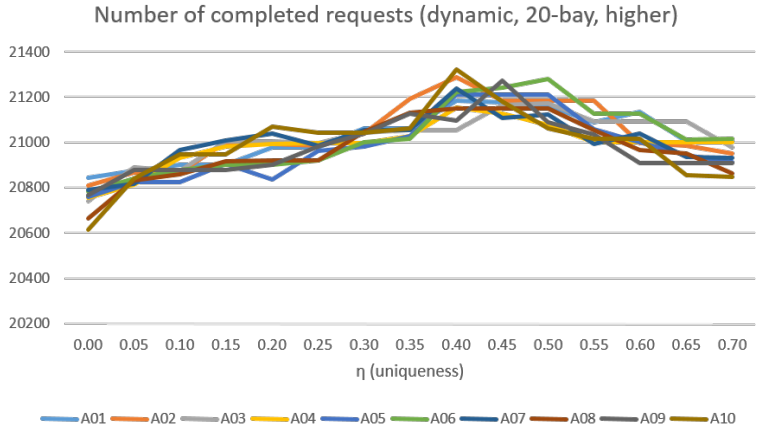
(a) Time in system



(b) Delay ratio



(c) Speed index



(d) Number of completed requests

Figure 31: Higher workload results (20-bay)

CHAPTER VI

CONCLUSIONS AND FUTURE WORKS

In this dissertation, we proposed a method for designing a track layout of unified AMHSs in semiconductor manufacturing. We assumed that the AMHS employed dynamic routing to dispatch vehicles based on continuously updated traffic conditions. To avoid congestion, the system could reroute the path of any vehicle. To reflect vehicle movements under dynamic routing, the track layout included rerouted traffic flows.

Focusing on a shortcut placement problem, we developed an optimization problem and combined it with simulation to determine the best shortcut placements. We formulated it as a multi-commodity network design problem where two types of flow variables fulfilled the transfer of each commodity: the flow over the shortest path, and the aggregated flow over alternative paths. We defined commodities and other input data based on the routing simulator. Using simulation, we validated the track layouts and calculated the average travel time of each edge and the rerouting frequency of each commodity. We developed a heuristic to obtain a solution from the iterations between optimization and simulation. Without exploring a large number of feasible solutions, the heuristic provided a solution design within a reasonable number of iterations. The computational results showed that the layout obtained by using our approach outperformed those obtained by using a classic multi-commodity network design problem and the scenario-based approaches presented in Chapter 3. We also observed a significant relationship between routing performance and the uniqueness of alternative paths.

We suggest three possible extensions of the research presented in this dissertation:

Design of a track layout incorporating the in-bay track design Because unified AMHSs support direct transfers between two machines, inter-bay traffic can also affect intra-bay. Although the base graph has edges corresponding to the longest lanes of a bay, in this dissertation, we simplified the bays and included them in the base graph. Clearly, the problem will have more commodities as more stopping locations are added.

Analytic model of routing performance based on layout characteristics Our computational results illustrated the relationship between the uniqueness of alternative paths and the performance of dynamic routing. Although we did not discuss a closed-form relation, our computational results merit further investigation.

Other applications Other applications of the network design problem with alternative paths may need to relax the assumptions regarding the characteristics of a track layout, such as unidirectional track segments and no parallel lanes between two locations. One potential application is a railway network consisting of parallel lanes and few intersections with many branches; trains change paths only at intersections.

APPENDIX A

CONSTRAINTS FOR A BASE TRACK

This chapter presents the constraints of NDPA, which describes a base track layout. Constraints for specific locations are grouped together. These constraints, however, need be modified when we apply NDPA to other areas.

$G = (N, E)$ has two edges with alternating directions between adjacent nodes. The optimization problem selects edges and determines their directions to represent a track layout. For each adjacent node pair $\{i, j\}$, $y_{(i,j)} = 1$ if edge (i, j) is chosen; 0 otherwise. None of node pairs satisfies both $y_{(i,j)} = 1$ and $y_{(j,i)} = 1$ together. As our optimization problem starts with a spine track layout, it must have constraints that define a base track layout. The constraints for a base track layout consist of multiple groups.

For notational convenience, let B denote the set of nodes for bays. Given $BW = BH = 3$, we have 9 nodes, $B_i(1), B_i(2), \ldots, B_i(9)$ for bay i.

- Upper bays: For bay $i = 1, \ldots, NB/2$, its first node is

$$B_i(1) = (NO + 1) \times (\text{Column}) - (NO + 2) - (3 + BB) \times (i - 1),$$

and the other nodes are

$$\begin{aligned}
B_i(j) &= B_i(j-1) - 1 && j = 2, 3, \\
B_i(j) &= B_i(j-1) + (\text{Column}), && j = 4, 5 \\
B_i(j) &= B_i(j-1) + 1, && j = 6, 7, \\
B_i(8) &= B_i(7) - (\text{Column}), \\
B_i(9) &= B_i(8) - 1.
\end{aligned}$$

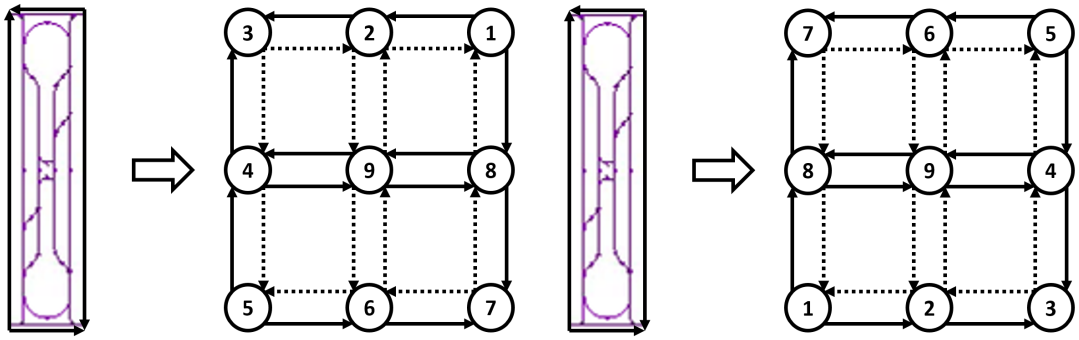
- Lower bays: For bay $i = 1, \ldots, NB/2$, its first node is

$$B_{i+NB/2}(1) = ((\text{Row}) - (NO+1)) \times (\text{Column}) + (NO+2) + 1 + (3+BB) \times (i-1),$$

and the others are defined as follows:

$$\begin{aligned}
B_{i+NB/2}(j) &= B_{i+NB/2}(j-1) + 1, && j = 2, 3, \\
B_{i+NB/2}(j) &= B_{i+NB/2}(j) - (\text{Column}), && j = 4, 5, \\
B_{i+NB/2}(j) &= B_{i+NB/2}(j-1) - 1, && j = 6, 7 \\
B_{i+NB/2}(8) &= B_{i+NB/2}(7) + (\text{Column}), \\
B_{i+NB/2}(9) &= B_{i+NB/2}(8) + 1.
\end{aligned}$$

Edges for bays have same directions. Simplified bays have four sides: the innermost outer loop lane, the outermost center loop lane, and two main lanes with stopping locations, the directions of which are fixed. We label bay nodes in order to distinguish edges. The first node is the entrance from the innermost outer loop lane. Note that it is not included in counting NO. Then, the nodes are labeled according to directions of edges.



(a) Upper bays (b) Lower bays

Figure 32: Grid representation of bays

We assume that a bay is simplified to a 3×3 square grid, i.e. $BW = BH = 3$. Figures 32(a) and 32(b) illustrate how to label bay nodes and which edges to be selected. Solid arrows will be chosen for a track layout.

We select four side lanes and four horizontal in-bay edges as above. Four vertical in-bay edges will not be chosen. We aggregate transfers with respect to bays, and the center node of each bay is considered as an origin or destination location. Although bay nodes are numbered differently in the lower area, we can apply the same constraints.

$$\text{Side 1 :} y_{B_i(1),B_i(2)} = 1,\ y_{B_i(2),B_i(3)} = 1,$$

$$\text{Side 2 :} y_{B_i(4),B_i(3)} = 1,\ y_{B_i(5),B_i(4)} = 1,$$

$$\text{Side 3 :} y_{B_i(5),B_i(6)} = 1,\ y_{B_i(6),B_i(7)} = 1,$$

$$\text{Side 4 :} y_{B_i(8),B_i(7)} = 1,\ y_{B_i(1),B_i(8)} = 1,$$

$$\text{In-bay, horizontal :} y_{B_i(4),B_i(9)} = 1,\ y_{B_i(9),B_i(4)} = 1,\ y_{B_i(9),B_i(8)} = 1,\ y_{B_i(8),B_i(9)} = 1,$$

$$\text{In-bay, vertical :} y_{B_i(2),B_i(9)} = 0,\ y_{B_i(9),B_i(2)} = 0,\ y_{B_i(9),B_i(6)} = 0,\ y_{B_i(6),B_i(9)} = 0.$$

Each lane on the outer loop should have a consistent direction. Each lane is either clockwise or counter-clockwise. Different lanes on the outer loop may have different directions if $DO = 1$. We define directions of the outermost lane first.

- Top: $\forall j \in \{1, \ldots, NO\}$, $\forall i \in \{1, \ldots, (\text{Column}) - (2j-1)\}$,

$$\text{if } DO = 0, \quad y_{s_T+i+1,s_T+i} = 1.$$

$$\text{if } DO = 1, \quad y_{s_T+i,s_T+i+1} = 0.5 \times (1 + (-1)^{NO-j}),$$

$$y_{s_T+i+1,s_T+i} = 0.5 \times (1 + (-1)^{NO-j+1}).$$

  where $s_T = (j-1) \times (\text{Column}) + (j-1)$.

- Bottom: $\forall j \in \{1, \ldots, NO\}$, $\forall i \in \{1, \ldots, (\text{Column}) - (2j-1)\}$,

$$\text{if } DO = 0, \quad y_{s_B+i,s_B+i+1} = 1,$$

$$\text{if } DO = 1, \quad y_{s_B+i,s_B+i+1} = 0.5 \times (1 + (-1)^{NO-j+1}),$$

$$y_{s_B+i+1,s_B+i} = 0.5 \times (1 + (-1)^{NO-j}),$$

  where $s_B = ((\text{Row}) - j) \times (\text{Column}) + (j-1)$.

- Left: $\forall j \in \{1, \ldots, NO\}$, $\forall i \in \{1, \ldots, (\text{Row}) - (2j-1)\}$,

  $$\begin{aligned} &\text{if } DO = 0, \quad y_{s_L+(i-1)\times(\text{Column}),s_L+i\times(\text{Column})} = 1, \\ &\text{if } DO = 1, \quad y_{s_L+(i-1)\times(\text{Column}),s_L+i\times(\text{Column})} = 0.5 \times (1 + (-1)^{NO-j+1}), \\ &\qquad\qquad\quad y_{s_L+i\times(\text{Column}),s_L+(i-1)\times(\text{Column})} = 0.5 \times (1 + (-1)^{NO-j}), \end{aligned}$$

  where $s_L = s_T = (j-1) \times (\text{Column}) + (j-1) + 1$.

- Right: $\forall j \in \{1, \ldots, NO\}$, $\forall i \in \{1, \ldots, (\text{Row}) - (2j-1)\}$,

  $$\begin{aligned} &\text{if } DO = 0, \quad y_{s_R+i\times(\text{Column}),s_R+(i-1)\times(\text{Column})} = 1, \\ &\text{if } DO = 1, \quad y_{s_R+(i-1)\times(\text{Column}),s_R+i\times(\text{Column})} = .5 \times (1 + (-1)^{NO-j}), \\ &\qquad\qquad\quad y_{s_R+i\times(\text{Column}),s_R+(i-1)\times(\text{Column})} = 0.5 \times (1 + (-1)^{NO-j+1}), \end{aligned}$$

  where $s_R = j \times (\text{Column}) - (j-1) + 1$.

Each lane on the center loop should have a consistent direction. Each lane should be clockwise or counter-clockwise. Different lanes on the center loop may have different directions if $DC = 1$. We define directions of the outermost lane first.

- Top: When we count from the outermost lane, the j-th lane has $(\text{Column}) - 2\times(NO+1+(j-1)) = (\text{Column}) - 2\times(NO+j)$ nodes. $\forall j \in \{1, \ldots, NC\}$, $\forall i \in \{1, \ldots, (\text{Column}) - 2 \times (NO+j) - 1\}$,

  $$\begin{aligned} &\text{if } DO = 0, \quad y_{s_T+i,s_T+i+1} = 1, \\ &\text{if } DO = 1, \quad y_{s_T+i,s_T+i+1} = 0.5 \times (1 + (-1)^{j}), \\ &\qquad\qquad\quad y_{s_T+i+1,s_T+i} = 0.5 \times (1 + (-1)^{j-1}), \end{aligned}$$

  where $s_T = (\text{Column}) \times (NO + 2 + (j-1)) + NO + j$.

- Bottom: When we count from the outermost lane, the j-th lane has $(\text{Column}) - 2\times(NO+j)$ nodes. $\forall j \in \{1, \ldots, NC\}$, $\forall i \in \{1, \ldots, (\text{Column}) -$

$2 \times (NO + j) - 1\}$,

$$\text{if } DO = 0, \quad y_{s_B+i+1,s_B+i} = 1,$$
$$\text{if } DO = 1, \quad y_{s_B+i,s_B+i+1} = 0.5 \times (1 + (-1)^{j-1}),$$
$$y_{s_B+i+1,s_B+i} = 0.5 \times (1 + (-1)^{j}),$$

where $s_B = s_T + (\text{Column}) \times (2 \times (NC - j) + 1) = (\text{Column}) \times (NO + 2 + 2 \times NC - j) + NO + j$.

- Left: When we count from the outermost lane, the j-th lane has $2 \times (NC - (j-1))$ nodes. $\forall j \in \{1, \ldots, NC\}$, $\forall i \in \{1, \ldots, 2 \times (NC - (j-1)) - 1\}$,

$$\text{if } DO = 0, \quad y_{s_L+i\times(\text{Column}),s_L+(i-1)\times(\text{Column})} = 1,$$
$$\text{if } DO = 1, \quad y_{s_L+(i-1)\times(\text{Column}),s_L+i\times(\text{Column})} = 0.5 \times (1 + (-1)^{j-1}),$$
$$y_{s_L+i\times(\text{Column}),s_L+(i-1)\times(\text{Column})} = 0.5 \times (1 + (-1)^{j}),$$

  where $s_L = s_T = (\text{Column}) \times (NO + 2 + (j-1)) + NO + j$.

- Right: When we count from the outermost lane, the j-th lane has $2 \times (NC - (j-1))$ nodes. $\forall j \in \{1, \ldots, NC\}$, $\forall i \in \{1, \ldots, 2 \times (NC - (j-1)) - 1\}$,

$$\text{if } DO = 0, \quad y_{s_R+(i-1)\times(\text{Column}),s_R+i\times(\text{Column})} = 1,$$
$$\text{if } DO = 1, \quad y_{s_R+(i-1)\times(\text{Column}),s_R+i\times(\text{Column})} = 0.5 \times (1 + (-1)^{j}),$$
$$y_{s_R+i\times(\text{Column}),s_R+(i-1)\times(\text{Column})} = 0.5 \times (1 + (-1)^{j-1}),$$

  where $s_R = (\text{Column}) \times (NO + 2 + j) - (NO + 1 + (j-1))$

Bays in the upper area are connected, so are bays in the lower area. There are two track segments between two adjacent bays, which we call top and bottom lanes. Indeed, top and bottom lanes are parts of the innermost outer loop lane or the outermost center loop lane, respectively. However, we treat them as connecting lanes between two bays. In addition, we prevent the problem from selecting two edges between top and bottom lanes.

Given NB, we have $NB/2-1$ lanes, and each lane consists of $BB+1$ edges. Two lanes are defined between bays 1 and 2. One is the lane from the upper exit of bay 1 to the upper entrance of bay 2, i.e., from $B_1(3)$ to $B_2(1)$ are connected. The other starts from the lower exit of bay 2 to the lower entrance of bay 1, i.e. from $B_2(7)$ to $B_1(5)$. $2 \times (BB+1)$ edges between those two lanes, i.e. edges between $B_1(4)$ and $B_2(8)$ should not be chosen. By definition, we have $B_2(1) = B_1(3) - (BB+1)$, $B_2(7) = B_1(5) - (BB+1)$, and $B_2(8) = B_1(4) - (BB+1)$. Therefore, for $i = 1, \ldots, NB/2-1$, for $j = 1, \ldots, BB+1$,

$$y_{B_i(3)-(j-1),B_i(3)-j} = 1, \quad y_{B_i(5)-j,B_i(5)-(j-1)} = 1,$$
$$y_{B_i(4)-j,B_i(4)-(j-1)} = 0, \quad y_{B_i(4)-(j-1),B_i(4)-j} = 0.$$

Connecting lanes for lower bays are defined in the same way, but bay $NB/2+1$ is placed to the left of $NB/2+2$. For $i = NB/2+1, \ldots, NB-1$, for $j = 1, \ldots, BB+1$,

$$y_{B_i(1)+j,B_i(1)+(j-1)} = 1, \quad y_{B_i(7)+(j-1),B_i(1)+j} = 1,$$
$$y_{B_i(8)+(j-1),B_i(1)+j} = 0, \quad y_{B_i(8)+j,B_i(1)+(j-1)} = 0.$$

The outer and center loops are connected. We define four track segments: two from the outer loop to the center loop and the other two from the center loop to the outer loop.

$$y_{(NO+2)\times(\text{Column})+(NO+1),(NO+2)\times(\text{Column})+(NO+2)} = 1,$$
$$y_{(NO+6)\times(\text{Column})-1,(NO+6)\times(\text{Column})-2} = 1,$$
$$y_{(NO+3)\times(\text{Column})-2,(NO+3)\times(\text{Column})-1} = 1,$$
$$y_{(NO+5)\times(\text{Column})+(NO+2),(NO+5)\times(\text{Column})+(NO+1)} = 1.$$

Upper and lower bays are connected. We define track segments that connect upper and lower bays except the center loop. Bays $NB/2$ and $NB/2+1$ are

connected in the left side while bays 1 and NB are linked in the right side. From the viewpoint of an actual track layout, they consist of a. the innermost outer loop lane, b. the outermost center loop lane, c. predefined shortcut lanes in the outer loop, and d. the track segments connecting the center and the outer loops. Note that the first two are considered in a different manner in our problem.

a. Just inside the outer loop lanes, there are two vertical lanes that connect upper and lower bays. In fact, they are a part of the innermost outer loop lane, but we separate them from the other outer loop lanes as stated above. The left lane starts from the upper to lower areas while the right lane has the opposite direction.

   – Left: It starts from node $NO \times (\text{Column}) + NO + 3$ and moves left to $NO \times (\text{Column}) + NO + 1$. Then, it goes down to $((\text{Row}) - (NO + 1)) \times (\text{Column}) + NO + 1 = (NO + 5 + 2 \times NC) \times (\text{Column}) + (NO + 1)$. Finally, it moves right to $(NO + 5 + 2 \times NC) \times (\text{Column}) + (NO + 1) + 2$. Hence we need to define $2 + (5 + 2 \times NC) + 2$ edges.

$$
\begin{aligned}
y_{LIC(1)+2,LIC(1)+1} &= 1, \\
y_{LIC(1)+1,LIC(1)} &= 1, \\
y_{LIC(i),LIC(i+1)} &= 1, \quad \forall i \in \{1, \ldots, 5 + 2 \times NC\}, \\
y_{LIC(6+2\times NC),LIC(6+2\times NC)+1} &= 1, \\
y_{LIC(6+2\times NC)+1,LIC(6+2\times NC)+2} &= 1,
\end{aligned}
$$

   where $LIC(i) = (NO + i - 1) \times (\text{Column}) + NO + 1$.

   – Right: Similarly, we have $2 + (5 + 2 \times NC) + 2$ edges. The lane starts from $((\text{Row}) - NO) \times (\text{Column}) - NO - 2 = (6 + 2 \times NC + NO) \times (\text{Column}) - NO - 2$ and moves right to $(6 + 2 \times NC + NO) \times (\text{Column}) -$

NO. Then, it moves up to $(NO+1) \times (\text{Column}) - NO$. Finally, it goes left to $(NO+1) \times (\text{Column}) - NO - 2$.

$$
\begin{aligned}
y_{RIC(1)-2,RIC(1)-1} &= 1, \\
y_{RIC(1)-1,RIC(1)} &= 1, \\
y_{RIC(i),RIC(i+1)} &= 1, \quad \forall i \in \{1, \ldots, 5 + 2 \times NC\}, \\
y_{RIC(6+2\times NC),RIC(6+2\times NC)-1} &= 1, \\
y_{RIC(6+2\times NC)-1,RIC(6+2\times NC)-2} &= 1,
\end{aligned}
$$

where $RIC(i) = (6 + 2 \times NC + NO - (i-1)) \times (\text{Column}) - NO$.

b. We define two track segments: the left one starts at the upper exit of bay $NB/2+1$ and reaches the lower entrance of bay $NB/2$, and the right one is from the lower exit of bay 1 to the upper entrance of bay NB.

We define four edges: one from the lower exit of bay 1, one to the lower entrance of bay $NB/2$, one from the top exit of bay $NB/2+1$, and one to the top entrance of bay NB.

- Left: It starts from node $B_{NB/2+1}(7)$ to the left by one edge and goes up to $B_{NB/2}(5)-1$. Then, it turns right and goes to $B_{NB/2}(5)$. Hence we have

$$y_{B_{NB/2+1}(7),B_{NB/2+1}(7)-1} = 1, \quad y_{B_{NB/2}(5)-1,B_{NB/2}(5)} = 1,$$

and for $j = 1, \ldots, 2 \times NC + 1$,

$$y_{B_{NB/2+1}(7)-1-(j-1)\times(\text{Column}),B_{NB/2+1}(7)-1-j\times(\text{Column})} = 1.$$

- Right: It starts from node $B_1(7)$ to the right by one edge and goes down to $B_{NB}(5) - 1$. Then, it turns left and goes to $B_{NB}(5)$. Hence we have

$$y_{B_1(7),B_1(7)+1} = 1 \quad y_{B_{NB}(5)-1,B_{NB}(5)} = 1 = 1,$$

and for $j = 1, \ldots, 2 \times NC + 1$,

$$y_{B_1(7)+1-(j-1)\times(\text{Column}),B_1(7)+1-j\times(\text{Column})} = 1.$$

c. For compatibility with the layout generator, we define 8 shortcut lanes, 4 on the left and 4 on the right, in the outer loop.

$$y_{LIC(1),LIC(1)-1} = 1,$$
$$y_{LIC(2)-1,LIC(2)} = 1,$$
$$y_{LIC(5+2\times NC),LIC(5+2\times NC)-1} = 1,$$
$$y_{LIC(6+2\times NC)-1,LIC(6+2\times NC)} = 1,$$
$$y_{RIC(1)+1,RIC(1)} = 1,$$
$$y_{RIC(2),RIC(2)+1} = 1,$$
$$y_{RIC(5+2\times NC)+1,RIC(5+2\times NC)} = 1,$$
$$y_{RIC(6+2\times NC),RIC(6+2\times NC)+1} = 1.$$

d. We define four edges: one close to bay 1, one close to bay $NB/2$, one close to bay $NB/2 + 1$, and one close to bay NB.

$$y_{B_1(7)+1,B_1(7)+2} = 1,$$
$$y_{B_{NB/2}(5)-2,B_{NB/2}(5)-1} = 1,$$
$$y_{B_{NB/2+1}(7)+1,B_{NB/2+1}(7)+2} = 1,$$
$$y_{B_{NB}(5)-2,B_{NB}(5)-1} = 1.$$

Track segments are unidirectional. If an edge is chosen in one direction, then its other direction cannot be chosen, but there are exceptions. Two pairs of edges in each bay are selected together in order to use the center node as origin and destination locations,. Let E_B be the set of such edges. Since $BW = BH = 3$, we have

$$E_B = \bigcup_{i=1,\ldots,NB} \{(B_i(9), B_i(4)), (B_i(4), B_i(9)), (B_i(9), B_i(8)), (B_i(8), B_i(9))\},$$

and the constraints are

$$y_{(i,j)} + y_{(j,i)} \leq 1 \quad \forall i, j \in N : (i,j), (j,i) \in E \setminus E_B.$$

We can have only two types of intersections: merging or diverging. This condition requires the degree of all nodes to be no more than 3 with the exception of the center node of each bay.

$$\sum_{j:(i,j)\in E} y_{(i,j)} + \sum_{j:(j,i)\in E} y_{(j,i)} \leq 3 \quad \forall i \in N \setminus \{B_i(9) \mid i = 1, \ldots, NB\}.$$

REFERENCES

[1] Abraham, I., Delling, D., Goldberg, A. V., and Werneck, R. F., "Alternative routes in road networks," *ACM Journal of Experimental Algorithmics*, vol. 18, no. 1, pp. 1.3:1–1.3:17, 2013.

[2] Acar, Y., Kadipasaoglu, S. N., and Day, J. M., "Incorporating uncertainty in optimal decision making: Integrating mixed integer programming and simulation to solve combinatorial problems," *Computers & Industrial Engineering*, vol. 56, pp. 106–112, 2009.

[3] Agarwal, Y., "Design of survivable networks using three- and four-partition facets," *Operations Research*, vol. 61, no. 1, pp. 199–213, 2013.

[4] Agarwal, Y. K., "Polyhedral structure of the 4-node network design problem," *Networks*, vol. 54, no. 3, pp. 139–149, 2009.

[5] Agrawal, G. K. and Heragu, S. S., "A survey of automated material handling systems in 300-mm semiconductor fabs," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 1, pp. 112–120, 2006.

[6] Almeder, C., Preusser, M., and Hartl, R. F., "Simulation and optimization of supply chains: alternative or complementary approaches," *OR Spectrum*, vol. 31, no. 1, pp. 95–119, 2009.

[7] Andradóttir, S., "A review of simulation optimization techniques," in *Proceedings of the Winter Simulation Conference*, vol. 1, pp. 151–158, 1998.

[8] April, J., Glover, F., Kelly, J. P., and Laguna, M., "Practical introduction to simulation optimization," in *Proceedings of the Winter Simulation Conference*, vol. 1, pp. 71–78, 2003.

[9] Azadivar, F., "Simulation optimization methodologies," in *Proceedings of the Winter Simulation Conference*, vol. 1, pp. 93–100, 1999.

[10] Bader, R., Dees, J., Geisberger, R., and Sanders, P., "Alternative route graphs in road networks," in *Theory and Practice of Algorithms in (Computer) Systems* (Marchetti-Spaccamela, A. and Segal, M., eds.), vol. 6595 of *Lecture Notes in Computer Science*, pp. 21–32, Springer Berlin Heidelberg, 2011.

[11] Bahri, N., Reiss, J., and Doherty, B., "A comparison of unified vs. segregated automated material handling systems for 300 mm fabs," in *Proceedings ot the IEEE International Semiconductor Manufacturing Symposium*, pp. 3–6, 2001.

[12] Balakrishnan, A., Magnanti, T. L., Sokol, J. S., and Wang, Y., "Telecommunication link restoration planning with multiple facility types," *Annals of Operations Research*, vol. 106, no. 1-4, pp. 127–154, 2001.

[13] Balakrishnan, A., Magnanti, T. L., Sokol, J. S., and Wang, Y., "Spare-capacity assignment for line restoration using a single-facility type," *Operations Research*, vol. 50, no. 4, pp. 617–635, 2002.

[14] Balakrishnan, A., Mirchandani, P., and Natarajan, H. P., "Connectivity upgrade models for survivable network design," *Operations Research*, vol. 57, no. 1, pp. 170–186, 2009.

[15] Bartlett, K., *Congestion-aware dynamic routing in automated material handling systems*. PhD thesis, Georgia Institute of Technology, 2014.

[16] Bartlett, K., Lee, J., Ahmed, S., Nemhauser, G., Sokol, J., and Na, B., "Congestion-aware dynamic routing in automated material handling systems," *Computers & Industrial Engineering*, vol. 70, pp. 176–182, 2014.

[17] Bilgen, B. and Çelebi, Y., "Integrated production scheduling and distribution planning in dairy supply chain by hybrid modelling," *Annals of Operations Research*, vol. 211, no. 1, pp. 55–82, 2013.

[18] Binato, S., Pereira, M. V. F., and Granville, S., "A new Benders decomposition approach to solve power transmission network design problems," *IEEE Transactions on Power Systems*, vol. 16, no. 2, pp. 235–240, 2001.

[19] Bozer, Y. A. and kuan Yen, C., "Intelligent dispatching rules for trip-based material handling systems," *Journal of Manufacturing Systems*, vol. 15, no. 4, pp. 226–239, 1996.

[20] Byrne, M. D. and Bakir, M. A., "Production planning using a hybrid simulation - analytical approach," *International Journal of Production Economics*, vol. 59, no. 13, pp. 305–311, 1999.

[21] Calle, E., Urra, A., Marzo, J. L., Kuo, G.-S., and Guo, H.-B., "Minimum interference routing with fast protection," *IEEE Communications Magazine*, vol. 44, no. 10, pp. 104–111, 2006.

[22] Carson, Y. and Maria, A., "Simulation optimization: methods and applications," in *Proceedings of the Winter Simulation Conference*, pp. 118–126, 1997.

[23] Chandra, C. and Grabis, J., *Supply Chain Configuration: Concepts, Solutions, and Applications*, ch. Simulation modeling and hybrid approaches, pp. 173–195. New York, NY: Springer New York, 2016.

[24] Chen, J. C., Chen, C.-W., Tai, C.-Y., and Tyan, J. C., "Dynamic state-dependent dispatching for wafer fabrication," *International Journal of Production Research*, vol. 42, no. 21, pp. 4547–4562, 2004.

[25] Cho, H., Kumaran, T. K., and Wysk, R. A., "Graph-theoretic deadlock detection and resolution for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 413–421, 1995.

[26] Chujo, T., Komine, H., Miyazaki, K., Ogura, T., and Soejima, T., "Distributed self-healing network and its optimum spare-capacity assignment algorithm," *Electronics and Communications in Japan (Part I: Communications)*, vol. 74, no. 7, pp. 1–9, 1991.

[27] Chung, J. and Tanchoco, J. M. A., "Layout design with hexagonal floor plans and material flow patterns," *International Journal of Production Research*, vol. 48, no. 12, pp. 3407–3428, 2010.

[28] Cormican, K. J., Morton, D. P., and Wood, R. K., "Stochastic network interdiction," *Operations Research*, vol. 46, no. 2, pp. 184–197, 1998.

[29] Costa, A. M., "A survey on Benders decomposition applied to fixed-charge network design problems," *Computers & Operations Research*, vol. 32, no. 6, pp. 1429–1450, 2005.

[30] Crainic, T. G., "Service network design in freight transportation," *European Journal of Operational Research*, vol. 122, no. 2, pp. 272–288, 2000.

[31] Crainic, T. G. and Laporte, G., "Planning models for freight transportation," *European Journal of Operational Research*, vol. 97, no. 3, pp. 409–438, 1997.

[32] Dabbas, R. M., Chen, H.-N., Fowler, J. W., and Shunk, D., "A combined dispatching criteria approach to scheduling semiconductor manufacturing systems," *Computers & Industrial Engineering*, vol. 39, no. 3-4, pp. 307–324, 2001.

[33] Dabbas, R. M. and Fowler, J. W., "A new scheduling approach using combined dispatching criteria in wafer fabs," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 3, pp. 501–510, 2003.

[34] Dahl, G. and Stoer, M., "A cutting plane algorithm for multicommodity survivable network design problems," *INFORMS Journal on Computing*, vol. 10, no. 1, pp. 1–11, 1998.

[35] de Keizer, M., Haijema, R., Bloemhof, J. M., and van der Vorst, J. G. A. J., "Hybrid optimization and simulation to design a logistics network for distributing perishable products," *Computers & Industrial Engineering*, vol. 88, pp. 26–38, 2015.

[36] Delling, D., Goldberg, A. V., Pajor, T., and Werneck, R. F., "Customizable route planning in road networks," *Transportation Science*, 2015. Articles in Advance.

[37] Figueira, G. and Almada-Lobo, B., "Hybrid simulation-optimization methods: a taxonomy and discussion," *Simulation Modelling Practice and Theory*, vol. 46, pp. 118–134, 2014.

[38] Fu, M. C., Glover, F. W., and April, J., "Simulation optimization: a review, new developments, and applications," in *Proceedings of the Winter Simulation Conference*, pp. 83–95, 2005.

[39] Gaskins, R., Mariano, T., and Sparrow, M., "Dynamic traffic based routing algorithm," 2001. US Patent 6,285,951.

[40] Gavish, B., "Topological design of telecommunication networks-local access design methods," *Annals of Operations Research*, vol. 33, no. 1, pp. 17–71, 1991.

[41] Gnoni, M. G., Iavagnilio, R., Mossa, G., Mummolo, G., and Leva, A. D., "Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry," *International Journal of Production Economics*, vol. 85, no. 2, pp. 251–262, 2003.

[42] Govind, N., Roeder, T. M., and Schruben, L. W., "A simulation-based closed queueing network approximation of semiconductor automated material handling systems," *IEEE Transactions on Semiconductor Manufacturing*, vol. 24, no. 1, pp. 5–13, 2011.

[43] Grötschel, M., Monma, C. L., and Stoer, M., "Design of survivable networks," in *Network Models* (Meurant, G., ed.), vol. 7 of *Handbooks in Operations Research and Management Science*, ch. 10, pp. 617–672, Elsevier, 1995.

[44] Grover, W. D., Bilodeau, T. D., and Venables, B. D., "Near optimal spare capacity planning in a mesh restorable network," in *Proceedings of the IEEE Global Telecommunications Conference*, vol. 3, pp. 2007–2012, 1991.

[45] Henderson, S. G. and Mason, A. J., "Rostering by iterating integer programming and simulation," in *Proceedings of the Winter Simulation Conference*, vol. 1, pp. 1268–1275, 2003.

[46] Hsieh, C.-H., Cho, C., Yang, T., and Chang, T.-J., "Simulation study for a proposed segmented automated material handling system design for 300-mm semiconductor fabs," *Simulation Modelling Practice and Theory*, vol. 29, pp. 18–31, 2012.

[47] Huang, C.-W., Chen, H.-Y., Yu, R.-C., and Yu, C.-Y., "Method and system for smart vehicle route selection," 2008. US Patent 7,356,378.

[48] Im, K., Kim, K., Moon, Y., Park, T., and Lee, S., "The deadlock detection and resolution method for a unified transport system," *International Journal of Production Research*, vol. 48, no. 15, pp. 4423–4435, 2010.

[49] Kennington, J. L., Olinick, E. V., and Spiride, G., "Basic mathematical programming models for capacity allocation in mesh-based survivable networks," *Omega*, vol. 35, no. 6, pp. 629–644, 2007.

[50] Kerivin, H. and Mahjoub, A. R., "Design of survivable networks: a survey," *Networks*, vol. 46, no. 1, pp. 1–21, 2005.

[51] Kim, B. and Kim, S., "Extended model for a hybrid production planning approach," *International Journal of Production Economics*, vol. 73, no. 2, pp. 165–173, 2001.

[52] Kim, B.-I., Oh, S., Shin, J., Jung, M., Chae, J., and Lee, S., "Effectiveness of vehicle reassignment in a large-scale overhead hoist transport system," *International Journal of Production Research*, vol. 45, no. 4, pp. 789–802, 2007.

[53] Kim, B.-I., Shin, J., and Chae, J., "Simple blocking prevention for bay type path-based automated material handling systems," *The International Journal of Advanced Manufacturing Technology*, vol. 44, no. 7-8, pp. 809–816, 2009.

[54] Kim, B.-I., Shin, J., Jeong, S., and Koo, J., "Effective overhead hoist transport dispatching based on the hungarian algorithm for a large semiconductor fab," *International Journal of Production Research*, vol. 47, no. 10, pp. 2823–2834, 2009.

[55] Kobitzsch, M., "An alternative approach to alternative routes: HiDAR," in *Algorithms - ESA 2013* (Bodlaender, H. L. and Italiano, G. F., eds.), vol. 8125 of *Lecture Notes in Computer Science*, pp. 613–624, Springer Berlin Heidelberg, 2013.

[56] Kobitzsch, M., Radermacher, M., and Schieferdecker, D., "Evolution and evaluation of the penalty method for alternative graphs," in *Proceedings ot the 13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'13)*, vol. 33, pp. 94–107, 2013.

[57] Kuhl, M. E. and Laubisch, G. R., "A simulation study of dispatching rules and rework strategies in semiconductor manufacturing," in *Proceedings of the IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pp. 325–329, 2004.

[58] Kurosaki, R., Shimura, T., Komada, H., Kojima, T., and Watanabe, Y., "Low cost and short lead time amhs design using interbay/intrabay diverging and converging IMP method for 300 mm fab," in *Proceedings of the 9th International Symposium on Semiconductor Manufacturing*, pp. 48–51, 2000.

[59] Le-Anh, T. and de Koster, R. B. M., "On-line dispatching rules for vehicle-based internal transport systems," *International Journal of Production Research*, vol. 43, no. 8, pp. 1711–1728, 2005.

[60] Lee, S., Na, B., and Lee, J., "Development and applications of an AMHS congestion monitoring system in semiconductor manufacturing facilities." Submitted to SpringerPlus, 2015.

[61] Li, S., Tang, T., and Collins, D. W., "Minimum inventory variability schedule with applications in semiconductor fabrication," *IEEE Transactions on Semiconductor Manufacturing*, vol. 9, no. 1, pp. 145–149, 1996.

[62] Lu, S. C. H., Ramaswamy, D., and Kumar, P. R., "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 3, pp. 374–388, 1994.

[63] Luxen, D. and Schieferdecker, D., "Candidate sets for alternative routes in road networks," *ACM Journal of Experimental Algorithmics*, vol. 19, no. 2, pp. 2.7:1–2.7:28, 2014.

[64] Mackulak, G. T. and Savory, P., "A simulation-based experiment for comparing AMHS performance in a semiconductor fabrication facility," *IEEE Transactions on Semiconductor Manufacturing*, vol. 14, no. 3, pp. 273–280, 2001.

[65] Magnanti, T. L. and Raghavan, S., "Strong formulations for network design problems with connectivity requirements," *Networks*, vol. 45, no. 2, pp. 61–79, 2005.

[66] Magnanti, T. L. and Wong, R. T., "Network design and transportation planning: models and algorithms," *Transportation Science*, vol. 18, no. 1, pp. 1–55, 1984.

[67] Mittler, M. and Schoemig, A. K., "Comparison of dispatching rules for semiconductor manufacturing using large facility models," in *Proceedings of the Winter Simulation Conference*, vol. 1, pp. 709–713, 1999.

[68] Moorthy, R. L., Hock-Guan, W., Wing-Cheong, N., and Chung-Piaw, T., "Cyclic deadlock prediction and avoidance for zone-controlled AGV system," *International Journal of Production Economics*, vol. 83, no. 3, pp. 309–324, 2003.

[69] Nazzal, D. and Bodner, D. A., "A simulation-based design framework for automated material handling systems in 300 mm fabrication facilities," in *Proceedings of the Winter Simulation Conference*, vol. 2, pp. 1351–1359, 2003.

[70] Nazzal, D. and El-Nashar, A., "Survey of research in modeling conveyor-based automated material handling systems in wafer fabs," in *Proceedings of the Winter Simulation Conference*, pp. 1781–1788, 2007.

[71] Nazzal, D., Johnson, A., Carlo, H. J., and Jimenez, J. A., "An analytical model for conveyor based AMHS in semiconductor wafer fabs," in *Proceedings of the Winter Simulation Conference*, 2008.

[72] Nazzal, D. and McGinnis, L. F., "Analytical approach to estimating AMHS performance in 300mm fabs," *International Journal of Production Research*, vol. 45, no. 3, pp. 571–590, 2007.

[73] Nazzal, D. and McGinnis, L. F., "Expected response times for closed-loop multivehicle AMHS," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 4, pp. 533–542, 2007.

[74] Nguyen, A.-T., Reiter, S., and Rigo, P., "A review on simulation-based optimization methods applied to building performance analysis," *Applied Energy*, vol. 113, pp. 1043–1058, 2014.

[75] Nolan, R. L. and Sovereign, M. G., "A recursive optimization and simulation approach to analysis with an application to transportation systems," *Management Science*, vol. 18, no. 12, pp. B676–B690, 1972.

[76] Paraskevopoulos, A. and Zaroliagis, C., "Improved alternative route planning," in *Proceedings of the 13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'13)*, vol. 33, pp. 108–122, 2013.

[77] Raghavan, S., *Formulations and algorithms for network design problems with connectivity requirements*. PhD thesis, Massachusetts Institute of Technology, 1995.

[78] Raghavan, S., "Low-connectivity network design on series-parallel graphs," *Networks*, vol. 43, no. 3, pp. 163–176, 2004.

[79] Randazzo, C. D. and Luna, H. P. L., "A comparison of optimal methods for local access uncapacitated network design," *Annals of Operations Research*, vol. 106, no. 1-4, pp. 263–286, 2001.

[80] Rani, D. and Moreira, M. M., "Simulationoptimization modeling: a survey and potential application in reservoir systems operation," *Water Resource Management*, vol. 24, no. 6, pp. 1107–1138, 2010.

[81] Rong-Hong, J., "Design of reliable networks," *Computers & Operations Research*, vol. 20, no. 1, pp. 25–34, 1993.

[82] Rose, O., "The shortest processing time first (SPTF) dispatch rule and some variants in semiconductor manufacturing," in *Proceedings of the Winter Simulation Conference*, vol. 2, pp. 1220–1224, 2001.

[83] Rose, O., "Some issues of the critical ratio dispatch rule in semiconductor manufacturing," in *Proceedings of the Winter Simulation Conference*, vol. 2, pp. 1401–1405, 2002.

[84] Sakauchi, H., Nishiiiura, Y., and Hasegawa, S., "A self-healing network with an economical spare-channel assignment," in *Proceedings of the IEEE Global Telecommunications Conference*, vol. 1, pp. 438–443, 1990.

[85] Sel, Ç. and Bilgen, B., "Hybrid simulation and MIP based heuristic algorithm for the production and distribution planning in the soft drink industry," *Journal of Manufacturing Systems*, vol. 33, no. 3, pp. 385–399, 2014.

[86] Shanthikumar, J. G. and Sargent, R. G., "A unifying view of hybrid simulation/analytic models and modeling," *Operations Research*, vol. 31, no. 6, pp. 1030–1052, 1983.

[87] Sherali, H. D. and Smith, E. P., "A global optimization approach to a water distribution network design problem," *Journal of Global Optimization*, vol. 11, no. 2, pp. 107–132, 1997.

[88] Smith, J. C., Lim, C., and Sudargho, F., "Survivable network design under optimal and heuristic interdiction scenarios," *Journal of Global Optimization*, vol. 38, no. 2, pp. 181–199, 2007.

[89] Soni, S. and Pirkul, H., "Design of survivable networks with connectivity requirements," *Telecommunication Systems*, vol. 20, no. 1-2, pp. 133–149, 2002.

[90] Sturm, R., Seidelmann, J., Dorner, J., and Reddig, K., "An approach to robust layout planning of AMHS," in *Proceedings of the Winter Simulation Conference*, vol. 2, pp. 1366–1372, 2003.

[91] Sun, C. C., Puig, V., and Cembrano, G., "Integrated simulation and optimization scheme of real-time large-scale water supply network: applied to catalunya case study," *Simulation*, vol. 91, no. 1, pp. 59–70, 2015.

[92] Swisher, J. R., Hyden, P. D., Jacobson, S. H., and Schruben, L. W., "A survey of simulation optimization techniques and procedures," in *Proceedings of the Winter Simulation Conference*, vol. 1, pp. 119–128, 2000.

[93] Tyan, J. C., Chen, J. C., and Wang, F.-K., "Development of a state-dependent dispatch rule using theory of constraints in near-real-world wafer fabrication," *Production Planning & Control*, vol. 13, no. 3, pp. 253–261, 2002.

[94] Urra, A., Calle, E., and Marzo, J. L., "Partial disjoint path for multilayer protection in GMPLS networks," in *Proceedings of the 5th International Workshop on Design of Reliable Communication Networks*, pp. 165–170, 2005.

[95] Urra, A., Calle, E., and Marzo, J. L., "Reliable services with fast protection in IP/MPLS over optical networks," *Journal of Optical Networking*, vol. 5, no. 11, pp. 870–880, 2006.

[96] Veerasamy, J., S.Venkatesant, and Shah, J. C., "Spare capacity assignment in telecom networks using path restoration," in *Proceedings of the 3rd International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp. 370–374, 1995.

[97] Wang, C.-N., "The improvement of lot delivery time in 450 mm semiconductor manufacturing," *Applied Mathematics & Information Sciences*, vol. 8, no. 6, pp. 2983–2990, 2014.

[98] Wang, L.-F. and Shi, L.-Y., "Simulation optimization: a review on theory and applications," *Acta Automatica Sinica*, vol. 39, no. 11, pp. 1957–1968, 2013.

[99] Wollmer, R., "Removing arcs from a network," *Operations Research*, vol. 12, no. 6, pp. 934–940, 1964.

[100] Wood, R. K., "Deterministic network interdiction," *Mathematical and Computer Modelling*, vol. 17, no. 2, pp. 1–18, 1993.

[101] woon Yoo, J., Sim, E.-S., Cao, C., and Park, J.-W., "An algorithm for deadlock avoidance in an AGV system," *The International Journal of Advanced Manufacturing Technology*, vol. 26, pp. 659–668, 2005.

[102] Wu, M.-C., Huang, Y. L., Chang, Y. C., and Yang, K. F., "Dispatching in semiconductor fabs with machine-dedication features," *The International Journal of Advanced Manufacturing Technology*, vol. 28, no. 9-10, pp. 978–984, 2006.

[103] Wysk, R. A., Yang, N.-S., and Joshi, S., "Resolution of deadlocks in flexible manufacturing systems: avoidance and recovery approaches," *Journal of Manufacturing Systems*, vol. 13, no. 2, pp. 128–138, 1994.

[104] Xie, C. and Allen, T. T., "Simulation and experimental design methods for job shop scheduling with material handling: a survey," *The International Journal of Advanced Manufacturing Technology*, vol. 80, no. 1-4, pp. 233–243, 2015.

[105] Yang, J.-W., Cheng, H.-C., Chiang, T.-C., and Fu, L.-C., "Multiobjective lot scheduling and dynamic OHT routing in a 300-mm wafer fab," in *Proceedings ot the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1608–1613, 2008.

[106] Yang, T. and Peters, B. A., "A spine layout design method for semiconductor fabrication facilities containing automated materialhandling systems," *International Journal of Operations & Production Management*, vol. 17, no. 5, pp. 490–501, 1997.

[107] Zafar, H., Harle, D., Andonovic, I., and Khawaja, Y., "Performance evaluation of shortest multipath source routing scheme," *IET Communications*, vol. 3, no. 5, pp. 700–713, 2009.